



TAMPEREEN TEKNILLINEN YLIOPISTO

**MIKAEL BILLING**  
**OPPIMISYMPÄRISTÖ ROBOTIIKAN JA ETÄOHJELMOINNIN**  
**OPETUKSEEN**

Diplomityö

Tarkastaja: professori Reijo Tuokko  
Tarkastaja ja aihe hyväksytty  
automaatio-, kone- ja materiaalitek-  
niikan tiedekuntaneuvoston  
kokouksessa 8. helmikuuta 2012

# TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Konetekniikan koulutusohjelma

**BILLING, MIKAEL:** Oppimisympäristö robotiikan ja etäohjelmoinnin opetukseen

Diplomityö, 104 sivua, 98 liitesivua

Huhtikuu 2012

Pääaine: Koneteollisuuden tuotantotekniikka

Tarkastaja: professori Reijo Tuokko

Avainsanat: Robotiikka, simulointi, etäohjelmointi

Teollisuustuotannon kehitys on menossa automaattisen tuotannon suuntaan. Robotiikka on isossa roolissa automaation lisäämisessä nykyaikaisessa teollisessa tuotannossa. Simulointien käyttö suunnittelussa, sekä etäohjelmoinnit osana toimivaa tuotantoa ovat pian arkipäivää. Tuotannon kehityksessä on tarve lyhentää suunnitteluaikoja ja tehdä tuotannosta tehokkaampaa. Lisääntyvä automatisointi tuo vaatimuksia myös työntekijöiden osaamistason nostamiseen. Tämän työn tavoitteena oli luoda konsepti, jonka avulla robotiikan ja sen etäohjelmoinnin oppiminen olisi tehokasta. Konseptia on tarkoitus käyttää niin oppilaitoksissa kuin yritysten henkilökunnan koulutuksessa. Työ on rajattu käsittelemään Fanuc- robotteja Fastsimu- etäohjelmistoympäristössä.

Tämä työ on kehitysprojekti, jossa käytetään hyväksi alan kirjallisuutta, tutkimusraportteja ja tieteellisiä artikkeleita. Tutkiva osa toteutettu kirjallisuustutkimuksena ja kehitysosiossa käytetään käytännönläheistä ratkaisumallia. Kirjallisuustutkimuksessa käydään läpi robottiautomaatiota, erityisesti jäysteenpoiston sovelluksia, sekä etäohjelmointia ja simulointia. Työssä käsitellään myös pedagogisia osa-alueita ja selvitetään ne tekijät, mitkä vähintään tulee olla hallinnassa etäohjelmointiympäristön käyttämiseksi. Työssä rakennetaan etäohjelmoinnin testiympäristö, jonka avulla varmistetaan konseptin toimivuus. Näiden pohjalta syntyy simulointiohjelmiston ja robotin käyttöohjeet sillä tasolla että luodut harjoitukset voidaan toteuttaa. Lisäksi opetuksen tueksi valmistettiin opetuslaite robotin sisäisen kinematiikan havainnollistamiseksi ja ymmärtämiseksi.

Työn lopuksi arvioidaan lopputulos ja esitetään kehitysehdotuksia ja havaittuja rajoitteita konseptista.

# ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Mechanical Engineering

**BILLING, MIKAEL:** Environment for Teaching robotics and offline programming

Master of Science Thesis, 104 pages, 98 Appendix pages

April 2012

Major: Production engineering

Examiner: Professor Reijo Tuokko

Keywords: Robotics, Offline programming, Simulation

Industrial production development is moving in the direction of the automatic production. Robotics is increasing the role of automation in the modern industrial production. Reducing the complexity of the design, as well as the use of offline programming as part of a production are soon common methods. There is a need in the Industrial development to reduce the design time and to make the production more efficient. The increasing automation also brings requirements to raise the knowledge level of the employees, as well. This work aims to create a concept that allows the learning of Robotics and offline programming to be effective. The concept is made to be used in educational establishments and enterprises in the staff training. The work is limited to the Fanuc-robots and Fastsimu-simulation software.

This thesis is a development project in which literature, research reports and scientific articles in the field are utilized, especially in the research part of the thesis. In the development part a hands-on-approach is used. The literature part deals with robotic automation, the area of deburring applications in particular, as well as the offline programming and simulation. The thesis also includes a pedagogical part where the objective is to identify the factors that have to be mastered in order to use offline programming environments. A test environment was constructed to ensure that the concept is working. On this basis, instructions for the simulation software and robot were drawn up so that exercises created can be carried out. Also a teaching device was made for the understanding of the kinematics.

Finally, the thesis is assessed in the light of the outcomes, development proposals are presented and restrictions to the concepts identified.

## ALKUSANAT

Tämä työ on tehty Tampereen teknillisen yliopiston opinnäytetyönä. Diplomityö on tehty Vaasan ammattikorkeakoululle, tarkoituksena kehittää robotiikan koulutusta ja yritysysteistyötä. Fastems Oy on antanut käyttöön etäohjelmointiohjelmiston ja teknisen tuen ympäristön luomisen avuksi.

Haluan kiittää työni tarkastajana toiminutta professori Reijo Tuokkoa, sekä Fastemsin puolelta Ari Lylynojaa avusta työn saattamiseksi opinnäytetyötä vastaavaan muotoon, niin työn rajauksen kuin sen sisällönkin suhteen. Lisäksi kiitän perhettäni kärsivällisyydestä ja tuesta opiskeluaikana.

Mustasaarella \_\_\_\_/\_\_\_\_ 2012

Mikael Billing

Richardintie 1  
65610 MUSTASAARI  
040 5912854

# SISÄLLYS

TIIVISTELMÄ .....	ii
ABSTRACT .....	iii
ALKUSANAT .....	iv
LYHENTEET .....	vii
1 JOHDANTO .....	1
1.1 Vaasan ammattikorkeakoulu Oy .....	1
1.2 Fastems Oy Ab .....	2
1.3 Työn tausta .....	2
1.4 Työn tavoitteet .....	3
1.5 Työn rakenne .....	3
2 TEOLLISUUSROBOTIIKKA .....	6
2.1 Robottien rakenteet .....	7
2.2 Kinematiikka .....	11
2.2.1. Siirtomatriisit .....	13
2.2.2. Denavit-Hartenberg- menetelmä .....	14
2.2.3. Kinemaattiset tehtävät .....	17
2.3 Robotin ohjausjärjestelmä .....	20
2.4 Käyttökohteet .....	21
2.4.1. Lastuava työstö .....	22
2.4.2. Jäysteenpoiston robotisointi .....	25
2.5 Robotti solun osana .....	28
2.6 Robottisolun suunnittelu .....	29
2.7 Työkalut .....	29
2.7.1. Jäysteenpoiston työkalut .....	31
2.7.2. Voimaohjaus jäysteenpoistossa .....	35
2.7.3. Jäysteenpoistosolun suunnitteluprosessi .....	38
2.8 Teollisuusrobotin ohjelmointi .....	42
2.8.1. Fanuc- teollisuusrobotin ohjelmointi .....	42
2.9 Etäohjelmointi ja simulointi .....	47
2.9.1. Robotin etäohjelmointi .....	49
2.9.2. Kalibrointi .....	49
2.9.3. Ohjelmistot .....	54
2.9.4. Fastsimu- etäohjelmointiohjelmisto .....	55
2.10 Robottiturvallisuus .....	58
3 ROBOTIIKAN OPETUS .....	60
3.1 Robotiikan opetusmenetelmät ja sisältö .....	60
3.2 Robotiikan peruselementit .....	62
3.3 Jäysteenpoiston peruselementit .....	63
4 OPPIMISYMPÄRISTÖ .....	64
4.1 Oppimisympäristön kuvaus .....	64

4.2	Kinematiikan opetuslaitteisto.....	65
4.2.1.	Laitteiston rakenne.....	65
4.2.2.	Kinemaattiset ratkaisut .....	67
4.3	Etäohjelmointiympäristö.....	74
4.3.1.	Testiympäristön luonti ja verifiointi .....	74
4.3.2.	Robottisolun valmistelu .....	75
4.3.3.	Simulointimallin luonti.....	77
4.4	Harjoitukset.....	88
4.4.1.	Peruskäyttöharjoitukset.....	88
4.4.2.	Etäohjelmointiharjoitukset.....	89
4.5	Robotin ja etäohjelmoinnin ohjeet.....	91
5	YHTEENVETO JA PÄÄTELMÄT .....	92
	Lähteet.....	94
	Liite 1 : Kinematiikan laskentataulukko	
	Liite 2 : Fanuc robotin yleisohje	
	Liite 3 : Fastsimu etäohjelmointi yleisohje	
	Liite 4 : Robotiikka perusharjoitukset	
	Liite 5 : Etäohjelmointi perusharjoitukset	

## LYHENTEET

CAD	Tietokoneavusteinen suunnittelu (Computer Aided Design)
CAM	Tietokoneavusteinen valmistus (Computer Aided Manufacturing)
OLP	Etäohjelmointi (Offline Programming)
TKP	Työkalupiste, Työkalukoordinaatiston origo, jonka paikka ja asento on ilmoitettu työkalulaippaan sidotun koordinaatiston ns. nollatyökalun suhteen.
TCP	Työkalupiste (Tool Center Point).
DOF	Vapausaste (Degree Of Freedom)





# 1 JOHDANTO

Seuraavissa kappaleissa käydään läpi opinnäytetyön taustat sekä diplomityön tavoitteet. Lisäksi esitetään kuvaus työn rakenteesta ja sen sisällöstä.

## 1.1 Vaasan ammattikorkeakoulu Oy

Vaasan ammattikorkeakoulu kouluttaa insinöörejä, tradenomeja, sairaanhoitajia, sosionomeja, terveydenhoitajia ja restonomeja suomen, ruotsin ja englannin kielellä. Vahvuuksina ovat energiateknologian ja kansainvälisen liiketoiminnan osaaminen kiinteässä yhteistyössä alueen työelämään. Opiskelijoita on 3 600, joista aikuisopiskelijoita on 700. Henkilöstöä on noin 300.

Vaasan ammattikorkeakouluun kuuluu kuusi yksikköä. Koulutusalayksiköt ovat tekniikka ja liikenne, liiketalous ja matkailu sekä sosiaali- ja terveysala. Tutkimus-, kehitys- ja innovaatiotoiminnan koordinoitua ja hallinnointia varten on TKI-yksikkö. Näiden lisäksi on vielä opintoasiain yksikkö, opetuksen kehittämissyksikkö sekä hallintoyksikkö.

Ammattikorkeakoulu tarjoaa opetusta kolmella kielellä kaikkiaan 17 koulutusohjelmassa. Ammattikorkeakoulussa suoritettava 3,5...4-vuotinen tutkinto (210...240 op) muodostuu ammattiteoreettisista opinnoista, koulutusosalalle soveltuvasta harjoittelusta sekä opinnäytetyöstä, joka tehdään pääsääntöisesti yhteistyössä alueen työelämän kanssa.

Kone- ja tuotantotekniikan koulutus tarjoaa monipuolisen väylän opiskella insinööriksi. Opintojen sisältö on suunniteltu vastaamaan Vaasan seudun tuotannollisen teollisuuden tarpeita. Painopistealueena ovat 3D-kone- ja laitesuunnittelu, tietokoneavusteinen valmistus sekä teollisuuden projektien johtaminen ja tuotantotalous. Opetuksessa hyödynnetään yrityselämän käytössä olevia ohjelmistoja koneensuunnittelussa, robotii-

kassa, valmistustekniikassa ja yritystoiminnan ohjauksessa. Tiivis yhteistyö työelämän kanssa tuo opiskelijoille harjoittelupaikkoja ja opinnäytetöiden aiheita, ja usein myös työpaikan valmistumisen jälkeen. Yhteistyösopimukset paikallisten vientiyritysten kanssa tarjoavat mahdollisuuden suorittaa opintoja myös ulkomailla yrityksen projekteissa. [Vamk 2012]

## **1.2 Fastems Oy Ab**

Fastems valmistaa joustavia valmistusjärjestelmiä (FMS) ja robottisoluja lastuavan työstön CNC-työstökoneiden automaatioon sekä tarjoaa automaatioon liittyviä palveluja. heillä on yhteistyötä kaikkien merkittävien metallintyöstökoneiden valmistajien kanssa. Fastems tuottaa automaattioratkaisuja näiden koneiden avulla, toimii siis ns. integraattorina.

Fastemsin palveluksessa on nykyään 400 työntekijää Euroopassa, USA:ssa ja Aasiassa. Pääkonttori sijaitsee Tampereella. Liikevaihto vuonna 2010 oli 48,9 miljoonaa euroa.

Fastemsin moduulirakenteiset laitteisto- ja ohjelmistoratkaisut koostuvat yksittäisestä koneesta kokonaiseen tehtaan tuotantojärjestelmään. Moduulirakenne mahdollistaa järjestelmien laajennettavuuden tarpeen vaatiessa. Yrityksen tavoitteena on parantaa asiakkaidensa kilpailukykyä automaation ja miehittämättömien tuotantomenetelmien avulla. Tavoitteena on pitää tuotanto käynnissä kaikkina vuoden 8 760 tunnin aikana. [Fastems 2012]

## **1.3 Työn tausta**

Tämä diplomityö on tehty Tampereen teknillisessä yliopistossa päättötyönä. Työ jakaantuu robotiikan tutkimukseen ja oppimisympäristön kehitykseen. Lisääntyvä automaatio edellyttää konepajatuotannolta jatkuvasti joustavampia menetelmiä. Teollisuusrobotit ovat mukana useimmissa automatisointiprojekteissa ja uudet tekniikat vievät robotteja uusille alueille. Nykyaikainen kappaletuotanto vaatii jokaisessa vaiheessa automaattista käsittelyä ja yksinkertaiset automatisoinnit ovatkin jo pitkään toteutettu

teollisuusroboteilla. Nyt on tarve automatisoida myös haasteellisimpia prosesseja, kuten jäysteenpoisto ja muut käsityövaiheet.

Fastems on kehittänyt jäysteenpoistoon ja viimeistelyyn erikoistuneen etäohjelmointiohjelmiston, Fastsimun. Sen pohjana ja kehityksen lähtökohtana on Visual Componentsin 3DSimulate- ohjelmistoalusta. Ohjelmiston tarkoituksena on helpottaa robotitiratojen ohjelmointia ja simulointia kappaleiden jäysteenpoistossa sekä viimeistelyssä. Etäohjelmointiohjelmistojen käyttö myös muualla kuin auto- ja lentokone-teollisuudessa tulee lisääntymään voimakkaasti lähivuosina. Etäohjelmointiympäristöt kehittyvät, niin käytettävyyden, kuin hinnankin suhteen ja etäohjelmoinnin, sekä simuloinnin hyödyt ovat käytettävissä myös pienemmän mittakaavan konepajoissa.

Vaasan ammattikorkeakoulu kouluttaa muun muassa konetekniikan insinöörejä, ja robotiikka on yksi tärkeä osa-alue opetussuunnitelmassa. Etäohjelmointiympäristöt tuovat uusia haasteita opetukseen ja siinä käytettäviin menetelmiin. On opittava uusi ohjelmisto ja lisäksi robotin konfiguraatio ja koordinaatistojen käsitteet tulevat olla etäohjelmoitaessa paremmin hallinnassa. Tätä taustaa vasten olisi tärkeää löytää ne menetelmät joilla uudet tekniikat pystytään omaksumaan ja ottamaan tehokkaasti käyttöön.

## **1.4 Työn tavoitteet**

Työn tavoitteena on kehittää Fastsimu- etäohjelmointiohjelmistoa hyödyntävä robotiikan oppimisympäristö. Tavoitteena on luoda konsepti, missä robottisolusta tehdään simulointimalli ja toimiva etäohjelmointiympäristö. Robottisolu voi olla olemassa oleva tai vasta suunnitteluvaiheessa. Työ sisältää tarvittavien perusharjoituksien ja menetelmien kehittelyn, sekä testiympäristön luomisen. Työn tuloksena saadaan konsepti, jonka avulla voidaan opiskella robotiikkaa sekä etäohjelmointia. Konseptin, sekä Fastsimun avulla voidaan rakentaa oppimis- / etäohjelmointiympäristö teollisuuden ja oppilaitosten tarpeisiin.

## **1.5 Työn rakenne**

Työ koostuu kolmesta osasta, joista ensimmäisessä perehdytään robotiikan teoriaan. Se on kirjallisuustutkimus teollisuusrobotiikasta. Siinä perehdytään erilaisiin robottirakenteisiin ja sovelluksiin. Robottikinematiikka koordinaatistoinen, sekä ulkoiset koor-

dinaatistot käydään läpi. Sovelluksista tutustutaan robotisoituun jäysteenpoistoon ja siinä käytettyihin menetelmiin ja työkaluihin. Myös robottien ohjelmointi ja simulointi käydään läpi.

Toisessa osassa tutustutaan robotiikan opetukseen ja siinä käytettäviin menetelmiin. Lisäksi etsitään ne peruselementit, joiden avulla robotin ohjelmointi ja simulointi onnistuu. Tämän pohjalta voidaan kehittää mahdollisimman yksinkertainen ja silti riittävän kattava konsepti aiheiden oppimiseksi.

Kolmannessa osassa kuvataan Oppimisympäristön kehitys. Menetelmäksi valittiin kokeellisen tuotekehityksen menetelmä. Kehitystyön taustalla on aiemmissa vaiheissa löydettyt osaamisvaatimukset robotiikkaan ja etäohjelmointiin. Oppimisympäristön kehitys jakaantuu kolmeen eri osioon: robotiikan teoria, peruskäyttö ja etäohjelmointi. Sama järjestys on robotiikan oppimisessa.

Robotiikan toiminnan perustana on robottikinematiikka, konfiguraatiot ja koordinaatistot. Robotin ohjausjärjestelmän tehtävä on siirtää varsiensa ja niveliensä avulla työkalulaippaan kiinnitettyä työkalua hallitusti määrättyihin paikoituksiin. Paikoituksella tarkoitetaan jäljempänä paikoituspistettä ja asemalla robotin työkalun sen hetkistä asemaa. Molemmat sisältävät tiedon paikasta ja asennosta kohdekoordinaatistossa. Työkaluja on hyvin monenlaisia kappaleenkäsittelyn tarraimista prosessityökaluihin.

Työkalua kuljetetaan robotin peruskoordinaatistossa ja useista varsista koostuva robotti pystyy toteuttamaan saman paikoituksen, eli työkalun paikan ja asennon, erilaisilla varsistojen asennoilla. Näitä eri asentoja kutsutaan konfiguraatioiksi. Työkalupisteen paikan ja tavoitepistettä vastaavat nivelkulmat ohjausjärjestelmä laskee nivelkulmiensa ja varsien pituuksien avulla. Menetelmää kutsutaan kinemaattiseksi tehtäväksi. Suorakulmaisia koordinaatistoja käytetään kuvaamaan peruskoordinaatiston ja käyttäjäkoordinaatistojen lisäksi myös paikoituksia ja työkaluja. Työkalukoordinaatisto ja paikoitus voidaan siis kuvata siirtomatriisien avulla peruskoordinaatistossa. Työkalukoordinaatisto liikkuu robotin liikkeiden mukana, kun paikoituskoordinaatistot pysyvät paikoillaan. Ohjausjärjestelmän tehtävä on ohjata robotin liikkeitä siten että koordinaatistot yhtyvät, eli paikoitus toteutuu. Järjestelmä on 6-akselisen nivelvarsirobotin yhteydessä monimutkainen koska varret ja nivelet muodostavat pitkän siirtomatriisien ketjun. Kuusi va-

pausastetta lisää myös paikoituksissa mahdollisien konfiguraatioiden määrää. Lisäksi järjestelmissä on useita työkaluja, paikoituksia ja käyttäjäkoordinaatistoja, joten kokonaisuus on melko monimutkainen. Järjestelmän hahmottamisen tueksi kehitettiin opetuslaite. 2-nivelisen tasomanipulaattorin avulla voidaan kinematiikka, konfiguraatit ja koordinaatistot esittää selkeämmin. Kehitetyn opetuslaitteen avulla voidaan osoittaa matemaattisten mallien toiminta käytännössä.

Robotin peruskäyttöharjoitukset koostuvat käsiohjaimella tapahtuvasta robottikäsi-varren liikuttamisesta ja ohjelmoinnista. Työkalujen ja koordinaatistojen määrittäminen on tärkeä osa myös käsiohjaimella ohjelmoitaessa, mutta etäohjelmoinnissa niiden ymmärtäminen on välttämätöntä.

Etäohjelmoinnin testiympäristön luonti ja verifiointi on oppimisympäristön kolmas osa. Testiympäristönä käytettiin Wärtsilän Technobothnia- laboratorioon sijoittamaa jäysteenpoistosolua. Solu koostuu lineaariradalle asennetusta Fanuc R-2000iB/165F 6-akselisesta teollisuusrobotista, jäysteenpoistotyökaluista, sekä Wärtsilä W-20 moottorin sylinterikannesta. Testiympäristöön valittiin solussa valmiina olevat harja-, Flap grinder- ja koneviilatyökalut. Solu mallinnettiin tarvittavin osin ja työkappaleiden avulla tehdyt robottiohjelmat testattiin oikeassa ympäristössä.

Harjoitustöiden ja käyttöohjeiden tekoa varten tehtiin toinenkin testiympäristö, min-kä avulla harjoitukset ja ohjeet testattiin. Ympäristö koostuu kuusiakselisesta Fanuc M-6iB nivelvarsirobotista ja kynätyökalusta. Työkappaleina käytettiin harjoituksia varten valmistetut muotoleikattu muovilevy ja kalteva sylinteripinta.

Työn lopussa on yhteenveto ja arviointi työn onnistumisesta, sekä päätelmät ja jatkokehitysehdotukset.

## 2 TEOLLISUUSROBOTIIKKA

Robotti on mekaaninen laite, joka voi tehdä osittain tai täysin tietokoneen ohjaamana mekaanisia toimenpiteitä. Standardin SFS-EN ISO 10218-1 mukaan Robotti on teollisuuden automaatiosovelluksissa käytettäväksi tarkoitettu automaattisesti ohjattu, uudelleen ohjelmoitavissa oleva monikäyttöinen käsittelylaite, jonka akseleista vähintään kolme on ohjelmoitavissa ja joka voi olla kiinteästi asennettu tai liikkuva [SFS-EN ISO 10218-1].

Robotit voidaan jakaa perustehtävän mukaisesti palvelurobotteihin ja teollisuusrobotteihin. Teollisuusrobotit toimivat nimensä mukaan teollisuudessa tehtaissa ja tuotantolaitoksissa ihmisten apuvälineinä, kun taas palvelurobotit toimivat ihmisen apuna erilaisissa ympäristöissä. Robotiikka ei ainoastaan käsitä robotteja, vaan pitää sisällään myös kaikki toimilaitteet ja anturit. Ilman näitä oheislaitteita robotti ei kykenisi hoitamaan sille tarkoitettuja tehtäviä. [Kuivanen 1999]

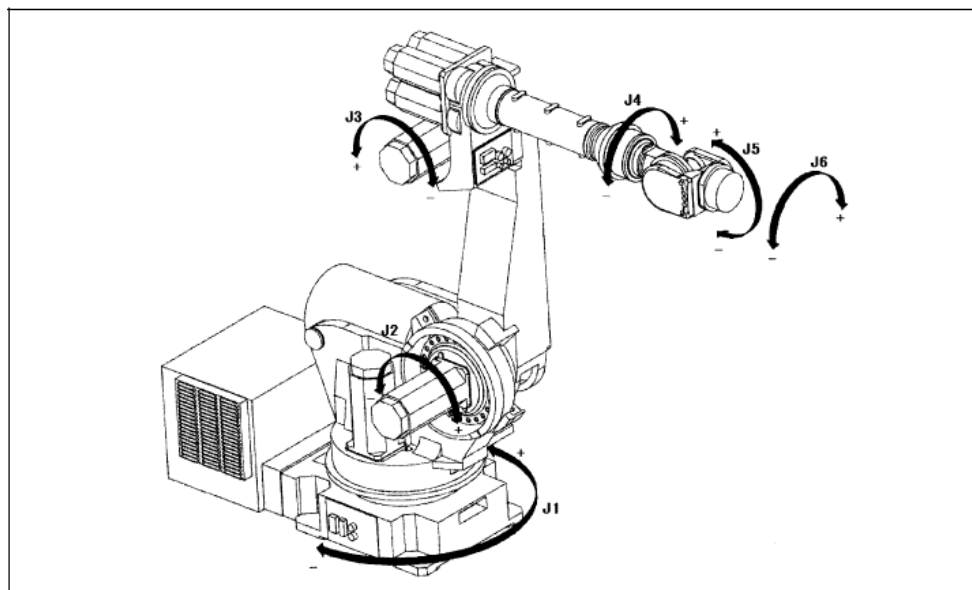
Teknologiateollisuus ry. on Trio Plus ohjelmassaan todennut että tuottavuuden nosto on merkittävä tekijä Suomen teollisuuden taistelussa globaaleja kilpailijoitaan vastaan. Tuottavuutta voidaan nostaa tutkimuksen mukaan kolmella tavalla: teknologisen kehityksen, työpanoksen laadun parantamisen ja pääomaintensiteetin kasvattamisen avulla. Robotiikka tuo osansa jokaiseen kolmeen tekijään, investoimalla uuteen ja joustavaan teknologiaan kasvattaa se samalla myös henkilöstön osaamistasoa. Nykyaikainen robotisoitu automaatiojärjestelmä pystyy tuottamaan tehokkaasti jopa yhden kappaleen sarjatuotantoa. [Teknologiateollisuus 2011]

Robotti- sana on saanut alkunsa Karel Čapekin tarinasta ”Rossum’s Universal Robots”, jossa Čapek kuvasi ihmismäisiä, sieluttomia ja tunteista vapaita koneita, jotka tottelivat ihmisisäntiänsä. Sana robotti sai siis alkunsa tshekin kielen sanasta robota, joka tarkoittaa yksitoikkoista työtä. Toisen maailmansodan jälkeen alettiin kehittää usei-

ta automaattisia koneita. Numeerisesti ohjatut koneet tulivat teollisuuteen 1950-luvulla. Koneiden ohjelmat olivat reikäkortteilla, kunnes vuonna 1956 George Devol kehitti magneettisen tallennus- ja toistolaitteen. Tietokoneiden kehityksen myötä kehitettiin ensimmäinen teollisuusrobotti. Ensimmäinen teollisuusrobotti, Unimation vihittiin General Motorsin käyttöön vuonna 1962. [Niku 2001]

## 2.1 Robottien rakenteet

Teollisuusrobottien perustehtävä on liikuttaa työkalulaippaansa kiinnitettyä työkalua hallitusti työalueellaan. Robotti koostuu useista tukivarsista, jotka liikkuvat toistensa suhteen joko jonkin suoran suuntaisesti (lineaariliike) tai suoran ympäri (kiertoliike). Näiden nivelien avulla tukivarret muuttavat keskinäisiä asentojaan ja asemiaan. Tätä robotin niveltä kutsutaan robotin vapausasteeksi (DOF, degree of freedom). Vapausasteiden määrä vaikuttaa robotin liikeratojen monipuolisuuteen. Kuvassa 1 on kuuden vapausasteen robotti, mikä pystyy paikoittamaan työkalulaipan työalueellaan vapaasti valittuun asentoon.



**Kuva 1.** Fanuc robotin kuusi vapausastetta J1...J6 [Fanuc 2001]

Yleisten nivelvarsirobottien rakenteesta löytyy ihmisen käsivarren, ranteen ja kämmenen liikkeitä vastaavat akselit. Rakenne ja robotin koko määrittelevät suurilta osin robotin kappaleenkäsittelykyvyn, ulottuman ja toistotarkkuuden. Nykyisissä roboteissa

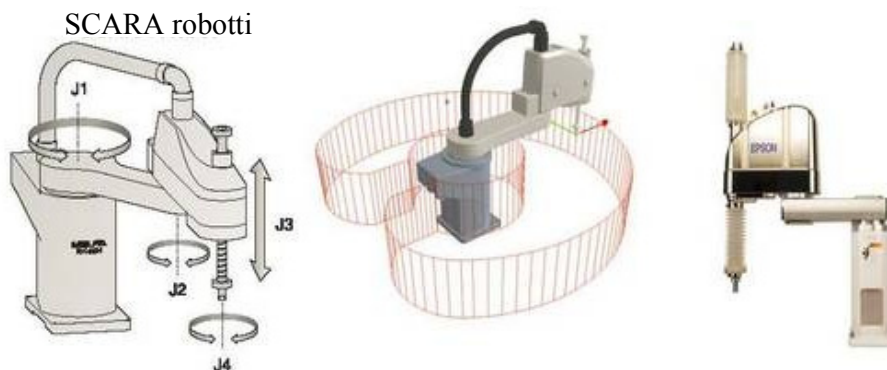
on yleensä kuusi tai neljä vapausastetta. Robotin perusrakenne tai työkappale voidaan myös kiinnittää liikkuvalla alustalla tai voidaan käyttää servo-ohjattuja työkaluja. Tällöin ohjattavia akseleita, eli vapausasteita voi olla robottisolussa kaksinkertainen määrä. Liikevarsia liikuttavat servomootorit ja jokaisessa nivelessä on anturointi kulman tai etäisyyden mittaamiseksi, kuvat 1 ja 6. [Niku 2001]

Suorakulmaisten robottien kolme ensimmäistä vapausastetta ovat lineaarisia. Suorakulmaisen rakenteen robotteja kutsutaan myös portaaliroboteiksi. Sen rakenne voi olla tuettu työalueen nurkista palkeilla tai rakenne on kiinnitetty palveltavan koneen runkoon. portaaliroboteja käytetään nosto- ja siirtotehtävissä. Kuormat voivat vaihdella muutamista kilogrammoista tuhansiin, kuva 2.



**Kuva 2.** Portaalirobotti ja työalue. [IFR 2012]

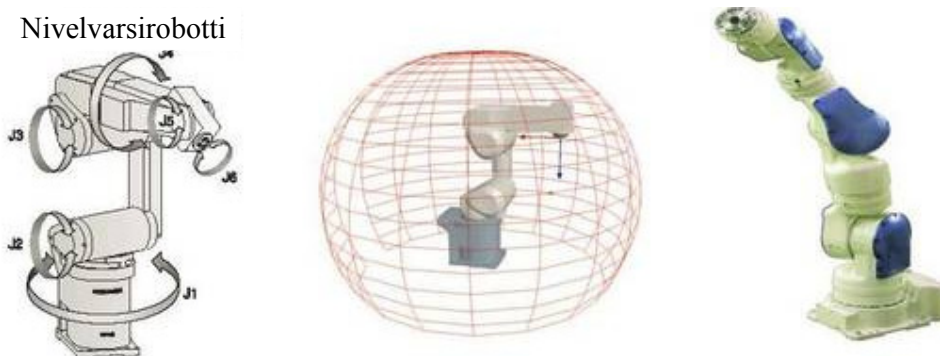
Scara - robotissa (Selective Compliance Assembly Robot Arm) on neljä vapausastetta, yksi pystysuuntainen lineaariakseli ja kolmella kiertyvillä nivelellä työkalu saadaan tasolla oikeaan kohtaan ja kiertymäkulmaan. Erittäin nopea robotti pienille, tyypillisesti alle 20 kg:n kuormille. Käytetään poiminta ja ladontatöissä, kuva 3.



**Kuva 3.** SCARA-robotti ja työalue. [IFR 2012]



Kiertyvänivelisessä robotissa kaikki vapausasteet ovat kiertyviä, kuva 4. Nämä ovat tavallisimpia teollisuusrobotteja. Vapausasteita on yleensä kuusi, viisi tai jopa seitsemän. Kiertyvänivelisien robottien koko määrää käsittelykyvyn ja työalueen halkaisijan. Käsittelykyky vaihtelee 1...2000 kg välillä. Työskentelyalueet ovat puolesta metristä aina viiteen metriin asti, kuva 4.



**Kuva 4.** Kiertyvänivelinen 6- ja 7-akselinen robotti. [IFR 2012]

Rinnakkaisrakenteisessa robotissa liikeakselit on kytketty rinnakkain. Tällaiset robotit ovat nivelvarsirobotteihin verrattuna jäykempiä rakenteellisesti, mutta niillä on kohdalaisen pieni työalue. Rinnakkaisrakenteisia robotteja valmistetaan kevytrakenteisina, kuva 5 vasen puoli, jolloin käsittelykyky on pienempi verrattuna oikean puolen raskasrakenteisen mallin käsittelykykyyn. Kevytrakenteisiksi tehtynä ne ovat nopeita ja tarkkoja poimintarobotteja ja raskasrakenteiset versiot hitaita, mutta erittäin tukevia ja soveltuvat jopa korvaamaan työstökoneita, kuva 5.



**Kuva 5.** Rinnakkaisrakenteisen robotin kaksi esimerkkiä. [Fanuc Robotics 2011]

Nykyiset teollisuusrobotit perustuvat lähes poikkeuksetta sarjakinemaattiseen rakenteeseen, jossa tukivarret on kytketty peräkkäin. Se mahdollistaa monipuoliset liikeradat ja jatkuvan kehityksen myötä myös lisääntyneen jäykkyyden ja tarkkuuden. Uutuutena sellaiset robottikäsi-robotit, joissa on yli kuusi vapausastetta monipuolisemman liikeradan saavuttamiseksi. Tällöin robotin on mahdollista työskennellä ahtaiden aukotuksien kautta työkohteisiin. [Kuivanen 1999].

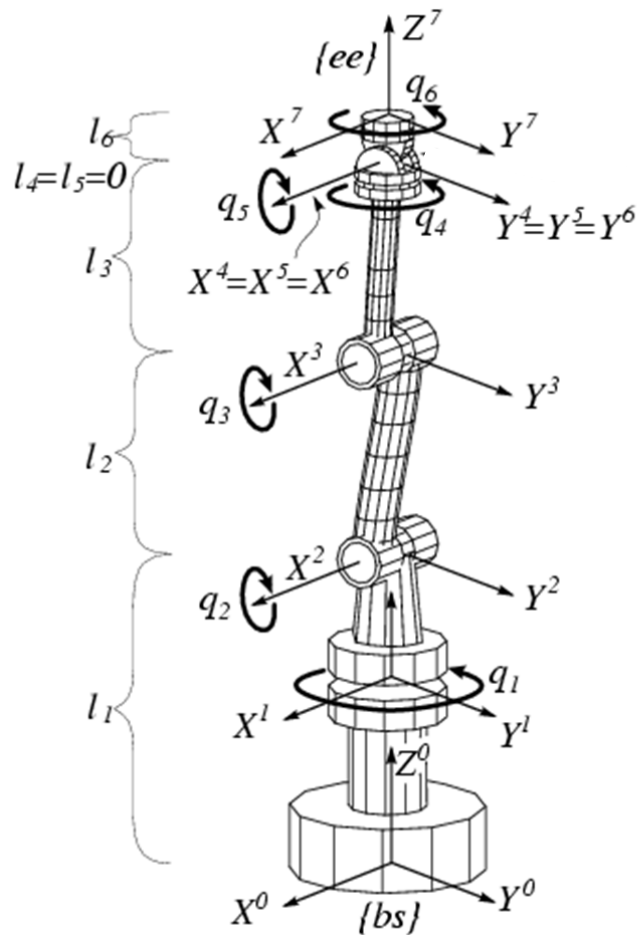


**Kuva 6.** Tyypillisiä teollisuusrobottirakenteita. [Fanuc Robotics 2011]

## 2.2 Kinematiikka

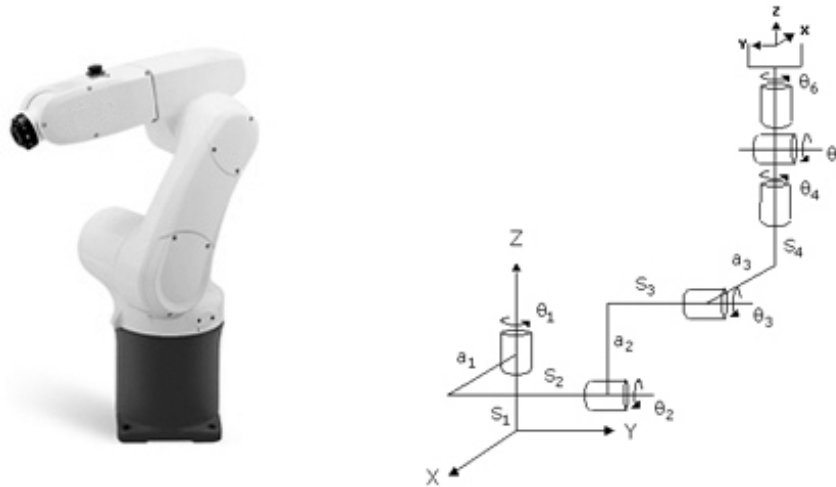
Kinematiikka tarkoittaa mekaanisten järjestelmien liikkeen analysointia huomioimatta kappaleisiin vaikuttavia voimia, kuten painovoima ja vääntömomentit. Se on matemaattinen lähestymistapa liikkeiden kuvaamiseksi. Kinematiikan avulla voidaan tutkia esimerkiksi asemia, nopeuksia ja kiihtyvyyksiä. Sitä käytetään niin robottien ohjauksissa, suunnittelussa ja ohjelmoinnin apuna. [Goldenberg 1999]

Robottikinematiikassa tarkastellaan mekanismin nivelpisteiden asemia, nopeuksia ja kiihtyvyyksiä. Kinemaattinen ketju koostuu toisiinsa liitetystä koordinaatistoilla kuvatuista nivelpisteistä. Ketjun alkupäässä on robotin peruskoordinaatisto ja kinemaattisen ketjun loppupäässä työkalupisteen koordinaatisto. Kuvassa 7 on tyypillinen 6-akselisen nivelvarsirobotin kinemaattinen ketju.



**Kuva 7.** 6-akselisen nivelvarsirobotin kinemaattinen ketju.  
[[http://www.orocos.org/kdl\\_old](http://www.orocos.org/kdl_old)]

Yleisin mekaaninen rakenne robotissa on sellainen, jossa tukivarsi aina kytketään edellisen perään (serial linked). Tätä rakennemuotoa kutsutaan avoimeksi kinemaattiseksi rakenteeksi



**Kuva 8.** Kuuden vapausasteen sarjakinemaattinen rakenne.  
[<http://www.preciseautomation.com>]

Kuvassa 8 näkyy 6-akselisen robotin kinemaattinen ketju ja siihen liittyvät parametrit, kuten nivelkulmat ja nivelvarsien dimensiot. Matemaattinen käsittely toteutetaan siirtomatriisien ja Denavit-Hartenberg laskentamenetelmän avulla. Ketjun viimeisen koordinaatiston paikka ja orientaatio voidaan kuvata peruskoordinaatistossa nivelkulmien funktiona. Toisaalta nivelkulmien avulla voidaan ratkaista ketjun viimeisen koordinaatiston paikka ja asento peruskoordinaatistossa. Ketjun viimeinen koordinaatisto on työkalupisteen koordinaatisto TKP.

### 2.2.1. Siirtomatriisit

Kahden koordinaatiston välinen paikka ja asento voidaan esittää homogeenisen siirtomatriisin avulla. Esimerkiksi merkintä  ${}^A_B\mathbf{T}$  tarkoittaa siirtomatriisia jonka avulla koordinaatisto B kuvataan koordinaatistoon A. Siirtomatriiseja käytetään yleensä robotin työalueen koordinaatistojen ja paikoituspisteiden kuvaamiseen robotin peruskoordinaatistoon.

$${}^A_B\mathbf{T} = \begin{bmatrix} {}^A_B\mathbf{A} & {}^B\mathbf{R} \\ 0 & 1 \end{bmatrix} \quad [1]$$

Siirtomatriisi koostuu 3x3 kiertomatriisista  ${}^A_B\mathbf{A}$ , mikä kuvaa koordinaatiston B orientaatiota kappaleen A suhteen, sekä 3x1 paikkavektorista  ${}^B\mathbf{R}$ , millä kuvataan koordinaatiston B:n paikka A koordinaatistossa.  ${}^B\mathbf{R}$  on muotoa

$${}^B\mathbf{R} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad [2]$$

Kiertomatriisi saadaan kertomalla jokaisen koordinaattiakselin yksikkövektorien kiertomatriisit keskenään. Laskusuunta riippuu kiertojärjestyksestä ja on sille käänteinen. Yhtälössä 3 on esitetty laskentakaava kun ensin on kierretty z-akselin ympäri sitten y- ja lopuksi x-akselin ympäri.

$${}^A_B\mathbf{A} = \mathbf{A}_x\mathbf{A}_y\mathbf{A}_z \quad [3]$$

Yhtälön 3 x-, y- ja z-akselien kiertomatriisit ovat seuraavat kun on valittu vastapäivään kierto positiiviseksi suunnaksi.

$$\mathbf{A}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \quad [4]$$

$$\mathbf{A}_y = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \quad [5]$$

$$A_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad [6]$$

Kiertomatriiseissa esiintyvät kulmat  $\gamma$ ,  $\varphi$  ja  $\theta$  kuvaavat koordinaatiston B koordinaattivektorien kulmaa suhteessa A koordinaatiston koordinaattivektoreihin. Esimerkiksi yhtälössä 4 on kuvattuna koordinaatiston A, y- ja z-akselien yksikkövektorit, B koordinaatiston y- ja z- akselien suuntaisina komponentteina. Toisin sanoen koordinaatistoa kierretään x-akselin ympäri.

Siirtomatriisiin käänteismatriisilla voidaan ratkaista koordinaatistomuunnos toiseen suuntaan, eli koordinaatisto A voidaan kuvata koordinaatistoon B seuraavasti:

$${}^B_A\mathbf{T} = ({}^A_B\mathbf{T})^{-1} \quad [7]$$

Robotin kinemaattinen ketju voidaan ilmaista siirtomatriisien avulla liittämällä jokaiseen niveleen koordinaatisto. Yhdistämällä kaikki ketjun siirtomatriisit voidaan ratkaista työkalulaipan paikka ja asento robotin peruskoordinaatistossa. Yhtälössä 8 on kuvattuna 6-nivelisen robotin siirtomatriisi.

$${}^0_6\mathbf{T} = {}^0_1\mathbf{T} \cdot {}^1_2\mathbf{T} \cdot {}^2_3\mathbf{T} \cdot {}^3_4\mathbf{T} \cdot {}^4_5\mathbf{T} \cdot {}^5_6\mathbf{T} \quad [8]$$

Robottiikassa käytetään koordinaatistoja ja siirtomatriiseja kuvaamaan robottisolun osien paikoituksia robotin peruskoordinaatistossa, kuten työkohteiden ja paikoituslaitteiden asemointia. Paikoituspisteitäkin käsitellään koordinaatistoina ja siirtomatriisien avulla robotti ohjaa työkalupisteen oikeaan paikkaan. Kappaleessa 2.3 on kuvattuna siirtomatriisien käyttöä robotin ohjauksessa.

### 2.2.2. Denavit-Hartenberg- menetelmä

Monimutkaisemmissa robottirakenteissa siirtomatriisien määrittäminen on haasteellista. Tiettyissä rakenteissa nivelen vaihtaessa asentoaan, muuttuu myös kaikki kolme niveleen liittyvän kiertomatriisin kulmaa. Tällaisissa tapauksissa jokaiseen niveleen liittyy siis kolme kulmamuuuttujaa, mikä vaikeuttaa automaattista käsittelyä. Määrittelyjä ja lasken-

taa helpottamaan on kehitetty useita menetelmiä, joista paljon käytetty on Denavit-Hartenberg menetelmä. Se perustuu siirtomatriisien ja nivelvarsiparametrien käyttöön. Menetelmä on käytössä robottien sisäisissä ohjauksissa, kuvaamaan työkalukoordinaatisto peruskoordinaatistoon.

Robottien niveliin kytkettyjen koordinaatistojen keskinäiset dimensiot voidaan esittää neljän parametrin avulla. Kaksi parametria kuvaa nivelvartta itseään ja kaksi muuta parametria sen kytkeytymistä vierekkäisiin varsiin. Menetelmässä jää vain yksi muuttuja, mikä on nivelen kulma tai lineaarinivelessä paikka. Tämä tekee laskemisesta huomattavan paljon helpompaa ja nivelien kulma-antureiden arvojen avulla voidaan matriisiyhtälöt ratkaista. Tätä menetelmää käytetään robotin sisäisen kinematiikan ratkaisussa, siis työkalupisteen paikan ja asennon kuvaamiseen peruskoordinaatistossa ja toisinpäin.

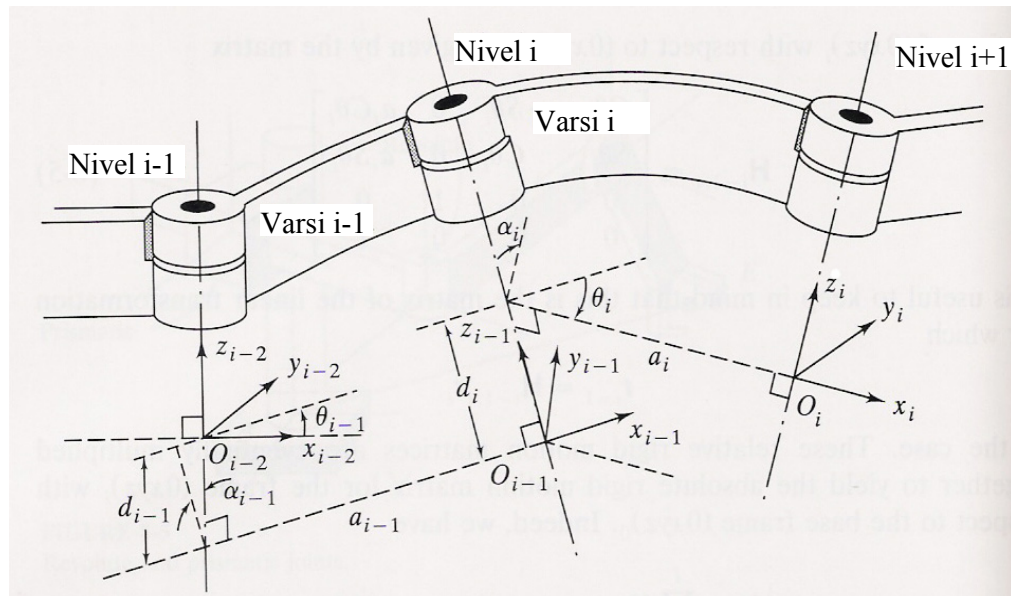
Kuvassa 9 olevat D-H parametrit seuraavasti:

$a_i$  = z-akselien välinen etäisyys edelliseen, koordinaatiston x-akselin suuntaan.

$\alpha_i$  = z-akselien välinen kulma edelliseen, koordinaatiston x-akselin suhteen

$d_i$  = x-akselien välinen etäisyys edellisen koordinaatiston z-akselin suhteen.

$\theta_i$  = z-akselien välinen kulma edellisen koordinaatiston z-akselin suhteen.



**Kuva 9.** Denavit-Hartenberg parametrit [Graig 2005]

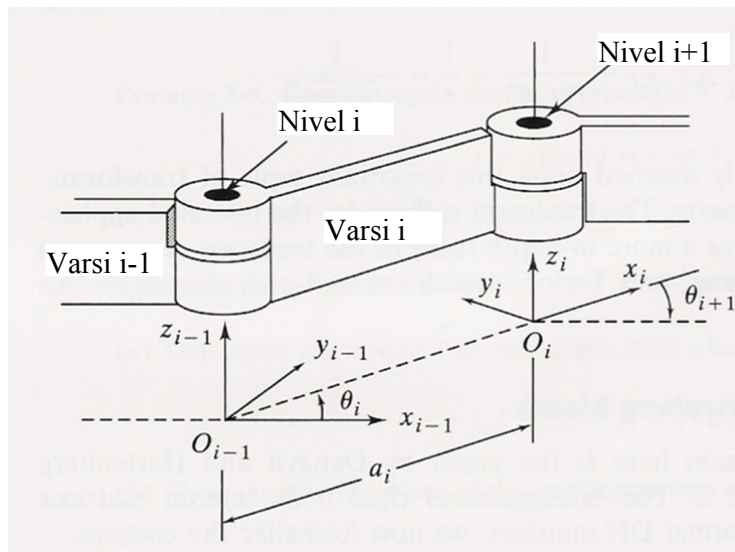
Parametrien avulla voidaan tarkastella monimutkaistakin mekanismia ja määrittää sen kinematiikka. Kuusinivelisen kiertonivelillä varustetun robotin kinematiikan esittämiseen tarvitaan 18 parametria, siis kuusi kertaa  $(a_i, \alpha_i, d_i)$ . Parametrit  $\theta_i$  ovat kulma-  
muuttujia, eli nivelien asentokulmia.

Yhtälössä 9 Denavit-Hartenberg siirtomatriisi, minkä avulla voidaan kuvata vierekkäisten nivelkoordinaatistojen asemia toistensa suhteen

$${}^{i-1}_iT = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [9]$$

Kun nämä kaikki siirtomatriisit kerrotaan keskenään saadaan homogeeninen siirtomatriisi, jonka avulla ketjun viimeinen koordinaatisto voidaan kuvata peruskoordinaatistoon. Menetelmä on sama kuin yhtälössä 1 homogeenisen siirtomatriisin tapauksessa.

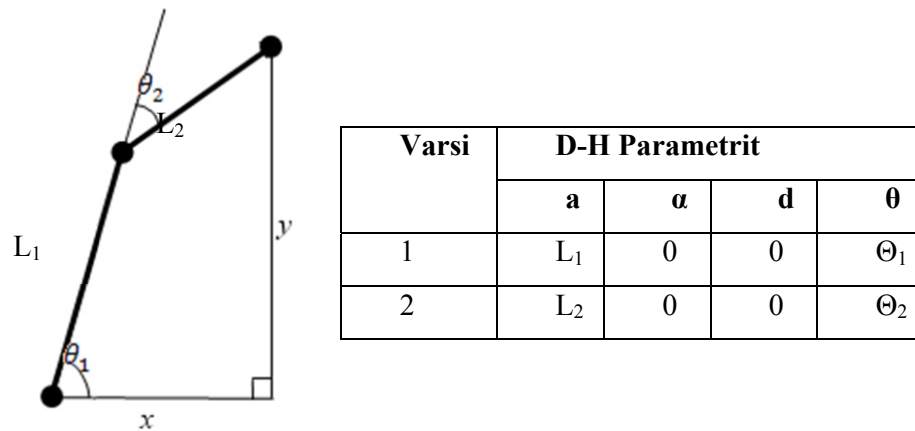
Tasomekanismin tapauksessa tilanne on helpompi, parametrit ovat esitettynä kuvassa 10. Siirtomatriiseista tulee yksinkertaisempia, koska kaksi neljästä varsiparametrilla ( $\alpha_i$  ja  $d_i$ ) ovat aina nollia.



**Kuva 10.** Denavit-Hartenberg parametrit tasotapauksessa [Graig 2005]



Kuvassa 11 on esitetty kahden vapausasteen tasomekanismi ja sen D-H parametrit.



**Kuva 11.** Denavit-Hartenberg parametrit 2-nivel tasomekanismissa.

Kappaleessa 4.2.2 käydään läpi 2-nivelisen tasomekanismin kinemaattiset ratkaisut geometrisesti ja D-H menetelmän avulla ratkaistuna.

### 2.2.3. Kinemaattiset tehtävät

Kinematiikan avulla ohjausjärjestelmä seuraa jatkuvasti TKP:n paikkaa sekä asentoa. Toisaalta se laskee tavoitepaikoitusten toteuttamiseen vaadittavia nivelkulma-arvoja jokaisen nivelen servo-ohjaimelle.

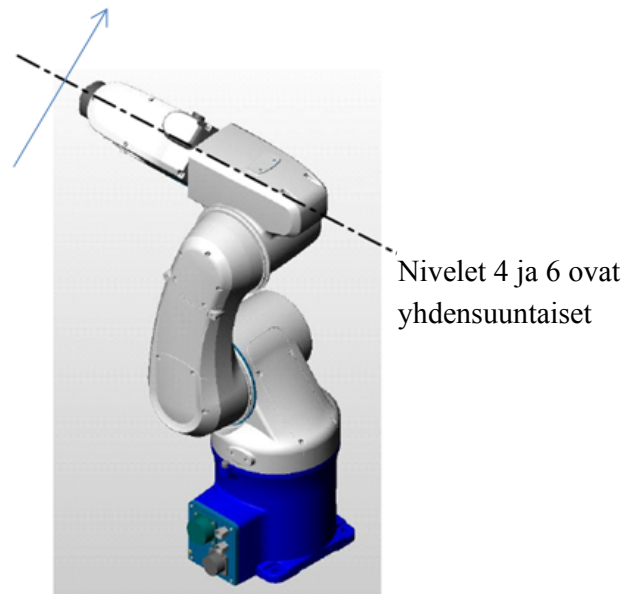
*Suora kinematiikka:* robotin tarraimen tai työkalun asema peruskoordinaatistossa lasketaan kinemaattisen ketjun eri vapausasteiden asemien ja kinemaattisen ketjun (liikevarsiston) tunnetun geometrian avulla. Suoran kinemaattisen tehtävän avulla ratkaistaan missä TKP- sijaitsee tällä hetkellä.

*Käänteinen kinematiikka:* Robotin tarraimen tai työkalun asema työavaruudessa on annettu. On siis olemassa paikoitus, mikä tulee toteuttaa. Tehtävänä on ratkaista vapausasteiden asennot ja konfiguraatio. Ratkaisu ei aina ole yksikäsitteinen, usein löytyy ratkaisu useammalla nivelkulmayhdistelmällä eli konfiguraatiolla. Kuvassa 12 on esitetty sama paikoitus toteutettuna kahdella eri konfiguraatiolla. Tällöin ohjelmoijan on valittava sopiva konfiguraatio.



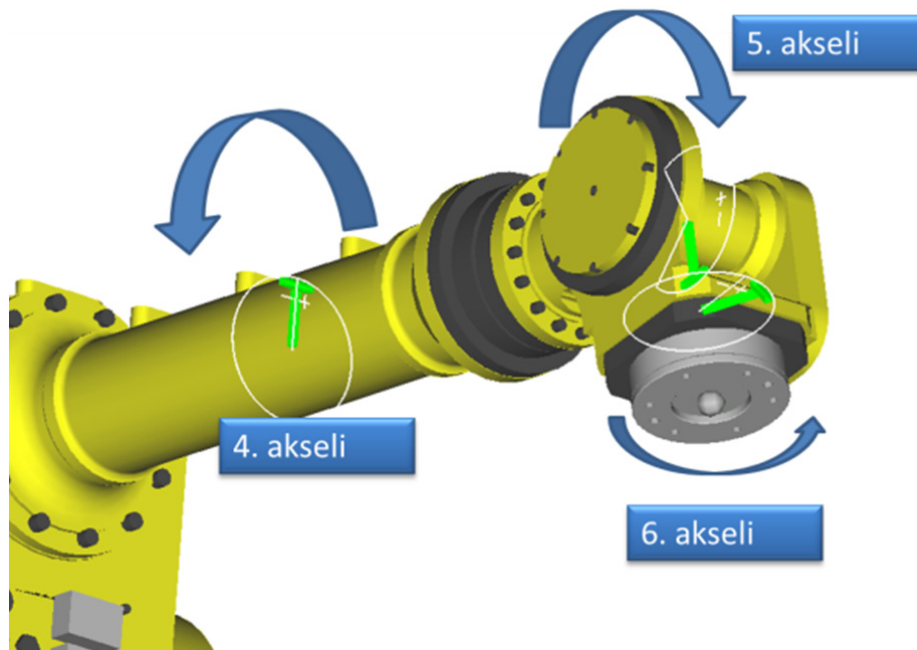
**Kuva 12.** Paikointus kahdella eri konfiguraatiolla. [<http://www.robotmaster.com>]

Konfiguraation lisäksi huomioonotettava seikka on myös singulariteetti. Tietyissä tilanteissa robotin kyky siirtää työkalupistettä valitussa asennossa katoaa. Ongelma esiintyy robotin ohjauksen suorittaessa paikoituksia lineaariliikkeenä ja kaksi akselia tulevat yhdensuuntaisiksi. Kuvassa 13 akselit 4 ja 6 ovat yhdensuuntaisia, lineaariliikkeen toteuttaminen ei onnistu, koska tilanteessa menetetään yksi vapausaste. Tällöin 4- ja 6-akseli ovat asennossa, jossa kiertonivelillä on yhteinen vapausaste. Ohjausjärjestelmät toimivat hieman eri tavoin singulariteettipisteissä. Yleensä robotti pysähtyy virhetilanteeseen. Käyttäjä voi kuitenkin sallia singulariteettipisteen läheisyydessä robotin poiketa interpoloidulta radalta ja ohittaa näin singulariteettipisteen. Nivelliikkeessä singulariteettiongelmaa ei ole, koska nivelet ohjautuvat itsenäisesti tavoiteasentoihin.



**Kuva 13.** Ranteen singulariteetti, nuoli osoittaa liikeradan. [<http://www1.adept.com>]

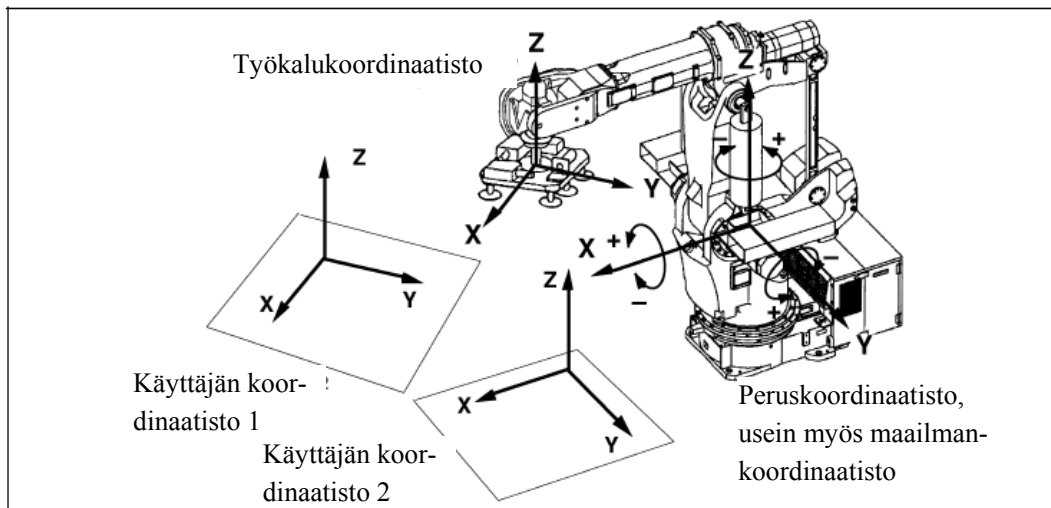
Singulariteetti voidaan korjata käyttämällä ongelmakohdissa nivelliikettä, muuttaa hie-  
man paikoitusta tai vaihtaa työkappaleen ja robotin keskinäistä asentoa. Hyvä yleissääntö 6-  
akselisen robotin käytössä on asentaa robotti niin, että normaalityössä 5. akseli ei olisi sa-  
mansuuntainen yläkäsivarren kanssa. Kuvassa 14 on esimerkki hyvästä asennosta.



**Kuva 14.** Singulariteetin välttävä työasento.

## 2.3 Robotin ohjausjärjestelmä

Robotin perustehtävä on liikutella työkalulaippaan kiinnitettyä työkalua tai tarrainta määrättyihin paikoituksiin. Tehtävän suorittamiseksi työkalupisteen paikka ja orientaatio l. asema on pystyttävä kuvaamaan matemaattisesti, jotta robotin ohjaus voidaan toteuttaa numeerisesti. Paikoitukset, jotka sisältävät asema- ja orientaatiotiedon esitetään kolmiulotteisessa koordinaatistossa. Kuvassa 15 on esitettyä Fanuc-robotin koordinaatistot. Yleisesti käytetään globaalia koordinaatistoa (World, Base), mikä sijoitetaan haluttuun kohtaan robotin rakenteessa. Lisäksi robotin työalueelle voidaan määrittää käyttäjän koordinaatistoja, mitkä helpottavat robotin ohjelmointia. [Graig 2005]

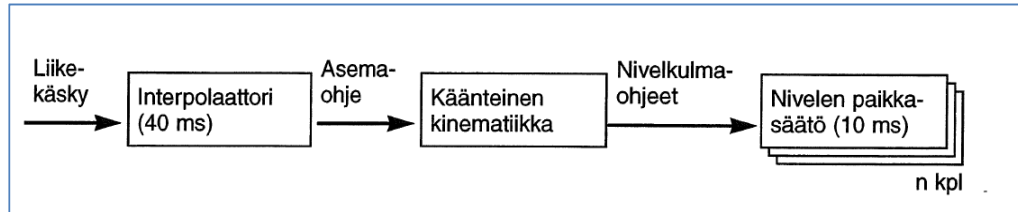


**Kuva 15.** Fanuc robotin koordinaatistot. [Fanuc 2001]

Työkalulaipassa on myös koordinaatisto, millä kuvataan työkalun paikka ja asento peruskoordinaatistossa. Työkalulaipassa on kiinteä koordinaatisto, jonka suhteen käyttäjä määrittelee käyttämänsä työkalun työkalupisteen.

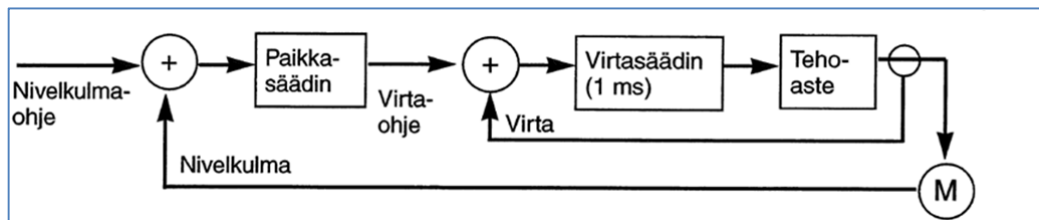
Kuvassa 16 ja 17 on esitettyä robotin ohjauksen toimintaperiaate liikekäskestä robotin paikoitukseen. Liikekäsky sisältää tavoitepisteen koordinaatit valitussa koordinaatistossa. Robotin ohjain tekee tarvittavat koordinaatistomuunnokset tavoitepisteen koordinaattien esittämiseksi robotin peruskoordinaatistossa. Tämän jälkeen Interpolaattori laskee nykypisteen ja tavoitepisteen välille tarvittavan määrän välipisteitä ohjel-

moidun radan toteuttamiseksi. Liiketapa on yleensä lineaari- tai ympyränkaari-interpolaatio.



**Kuva 16.** Robotin interpolointi ja kinematiikka liikeohjauksessa. [Kuivanen 1999].

Jos liikekäske tehdään nivelliikkeenä, ei TKP:n liikeradalla ole väliä, eikä interpolaatiota tarvita. Tällöin käänteinen kinematiikka ratkaisee suoraan tavoitepisteeseen vaadittavat nivelkulmaohjeet servo-ohjaukselle.



**Kuva 17.** Yksittäisen nivelen paikkasäätö. [Kuivanen 1999].

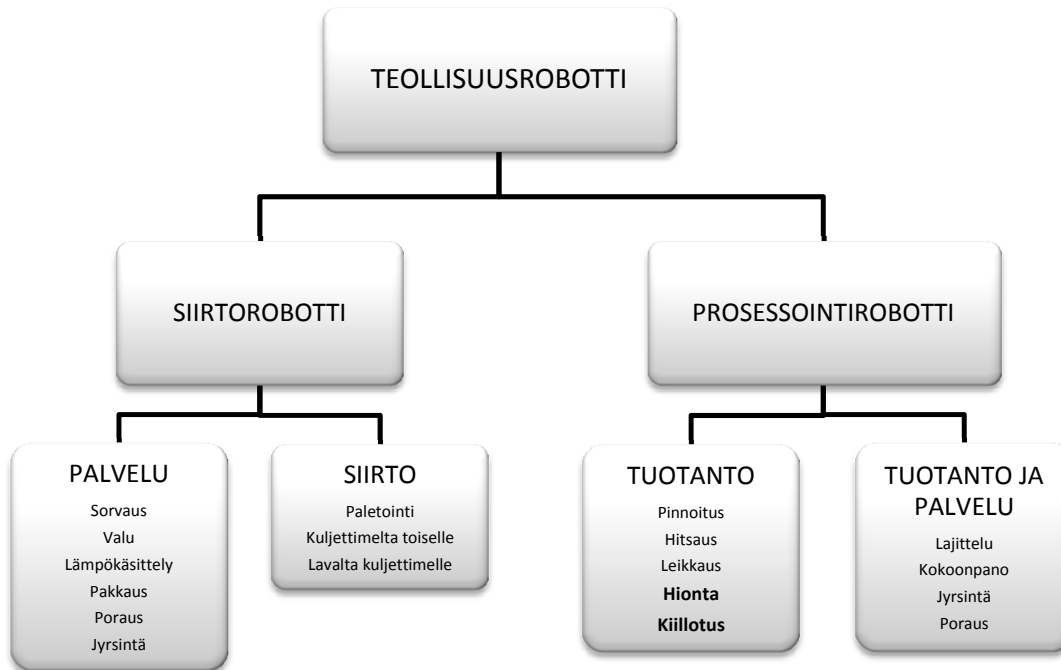
Paikkasäätö on takaisinkytketty järjestelmä, missä ohjearvona on nivelkulmaohje ja oloarvona nivelen fyysinen asento l. nivelkulma. Näiden erotuksena säädin syöttää sähkövirran servomotoorin virtasäätimelle, jossa on myös takaisinkytkentä virrasta, ja sitä kautta momentin tarkkailuun.

Ohjausjärjestelmä on erittäin nopea. Interpolaattori laskee uusia paikoituksia kymmeniä kertoja sekunnissa ja paikkasäädön ohjearvo päivittyy yli sata kertaa sekunnissa. Tämän monimutkaisen matemaattisen laskennan suorittaa robotin ohjausjärjestelmä. Käyttäjä voi keskittyä liikeratojen ja paikoitusten ohjelmointiin. [Kuivanen 1999]

## 2.4 Käyttökohteet

Teollisuusrobottien käyttökohteet ovat jakautuneet siirto- ja prosessointitehtäviin. Yleisesti ottaen robotit pystyvät tasaisempaan laatuun kuin ihminen, koska robotti pystyy

toistamaan tehtävänsä aina samanlaisena. Kuvassa 18 on esitettyä teollisuusrobotiikan yleiset tehtävät.



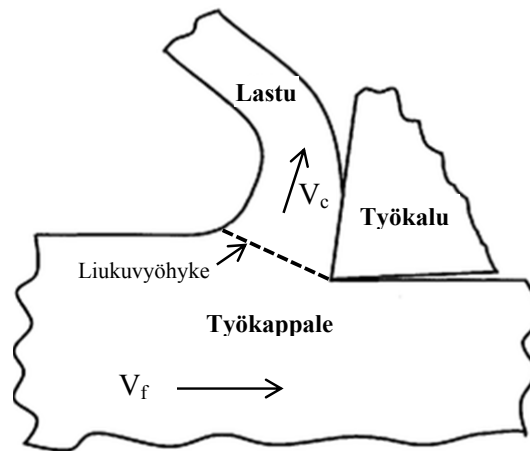
**Kuva 18.** Teollisuusrobotiikan tehtäväkenttä. [Kuivanen 1999]

Robottiikalla toteutettavat tehtävät lisääntyvät jatkuvasti robottien tekniikoiden kehittyessä. Robotit sopivat joustavina ja monipuolisina laitteina nykyaikaiseen nopeasti muutuvaan tuotantoon. [Niku 2001]

#### 2.4.1. Lastuava työ

Lastuavassa työssä irrotetaan työkalun avulla materiaalia työstettävästä kappaleesta, kuva 19. Työkalu tunkeutuu työkappaleeseen ja murtolujuuden ylittyessä liukuvyöhykkeellä lastu irtaantuu. Yleisimmät menetelmät ovat sorvaus, poraus ja jyrsintä. Lastuaviin menetelmiin kuuluvat myös hiomarakeilla hiovat menetelmät, kuten hiomakankaat ja harjat. Terällä lastuavissa menetelmissä tärkeä parametri on lastuamisnopeus (Velocity of Chip, Vc) [m/min], mikä on lastun irrotusnopeus. Lastuamisnopeus riippuu työstettävästä raaka-aineesta ja terän materiaalista sekä geometriasta. [David A. Stephenson, 1997]

Myös syöttönopeus on työkalun ja työstettävän kappaleen raaka-aineesta riippuva arvo, joka voidaan valita työkalutoimittajan suositustaulukosta. Pyörivillä työkaluilla tulee ottaa huomioon leikkaavien särmien lukumäärä syötön asettamiseksi särmäkohtaisesti. Syötön tunnus (Velocity of Feed,  $V_f$ ) [mm/r] tai [mm/min].



**Kuva 19.** Lastunmuodostuksen periaate [Milton C.Shaw, 2005]

Lastuamissyvyys yhdessä syötön kanssa määräävät lastun muodon ja poikkileikkauspinta-alan. Tärkeä parametri taloudellisesti tehokkaan työstön kannalta tarkasteltuna on lastuvirta [ $\text{cm}^3/\text{min}$ ]. Se kuvaa aineen poiston tehokkuutta. Lastuvirran optimointiin vaikuttavat koneen teho, tukevuus ja oikea lastunmuoto. Lastuamisarvojen valinta on usean tekijän summa. Työstettävä materiaali, työkalu, valittu menetelmä (viimeistely vai rouhinta), työstökone, yms. aiheuttavat oman erityispiirteensä työstöön.

### Suurnopeustyöstö

Suurnopeustyöstöllä tarkoitetaan lastuavia menetelmiä, joissa käytetään normaalia suurempia nopeuksia, tehoja tai lastuvirtoja. Menetelmiä ovat esimerkiksi jyrsintä, hionta ja sorvaus. Jokaisella menetelmällä on omat erityispiirteensä ja suurnopeustyöstön yhtenäistä määritelmää on vaikea löytää. Määrittelyä voidaan tehdä usealta eri näkökannalta, se on esimerkiksi

- työstää suurilla lastuamisnopeuksilla
- lastuamista nopeasti pyörivillä työkaluilla
- lastuamista nopealla syötöllä

- korkean lastuvirran työstöä.

Käytännössä suurnopeustyöstö tulisi ymmärtää prosessina, jossa käytetään tarkoitukseen soveltuvia tekniikoita ja menetelmiä. Ei siis pelkästään suuria kierroksia, vaan kokonaisvaltainen menetelmä suurilla lastuamisnopeuksilla ja tehokkaalla lastuvirralla. Voisi jopa sanoa että suurnopeustyöstöä voisi kuvata paremmin suuren tuottavuuden menetelmäksi, jossa kappale valmistuu rouhinnasta viimeistelyyn ja kiillotukseen. [Pasko, R. - Przybylski, L. & Slodki, B. 2002]

## **Suurnopeusjyrsintä**

Suurnopeusjyrsinnässä lastuaminen voi tapahtua suurella nopeudella. Esimerkiksi pyörimisnopeus voi olla 40 000...60 000 1/min ja syöttö 2000...3000 mm/min. Toisaalta tekniikoiden ja materiaalien kehittyessä suurnopeusjyrsinnän määrittely lastuamisarvojen mukaan muuttuu. Aiemmin suurnopeusjyrsinnäksi luokiteltu koneistus muuttuu perustekniikaksi ja suurnopeuden alue siirtyy. Suurnopeustyöstössä käytössä olevat lastuamissyvytykset ja -leveydet ovat pieniä. Lastuvirta on nopeuden ansiosta silti suuri. Koska pyörimisnopeudet ja kiihtyvyydet ovat suuria, niin käytettävät työkalut ovat pieniä, jopa 0,1mm halkaisijaltaan. Kun lastut ovat pieniä, niin saavutetaan erittäin pienet lastuamisvoimat. Tästä seurauksena on mittatarkka ja pinnanlaadultaan hyvä koneistusjälki. Koneistettava kappale ei myöskään lämpene, koska lämpö häviää lastujen mukana.

Robottien mekaaninen jäykkyys ja paikoitustarkkuus ovat yleisesti ottaen heikompia kuin työstökoneiden, mutta suurnopeusjyrsinnän menetelmiä hyväksikäyttäen robotitkin suoriutuvat osasta koneistuskeskuksilla tehtävistä töistä.

Tuottavuus ja kokonaiskustannukset riippuvat sekä työstöajasta ja asetusajasta. Suurnopeustyöstöllä voidaan pienentää työstöaikaa lisäämällä lastuvirtaa. Asetusaikaa voidaan lyhentää automaattisella kappaleen- ja työkalunvaihdolla, sekä etäohjelmoinnilla. Koneen älykkyyttä lisäämällä voidaan tehdä itseohjautuvia valmistussoluja, joissa tarkastukset tapahtuvat automaattisesti. Koska suurnopeustuotannossa vaaditaan koneilta hyvää ohjattavuutta ja luotettavuutta, niin monipuoliset koneet mahdollistavat automaation lisäämisen ja asetusajojen lyhenemisen. [Jawahir, I. S. 1994]



Parhaan mahdollisen hyödyn suurnopeuskoneesta saadaan kun käytetään oikeanlaisia työkaluja ja tehdään ohjelmat suurnopeustyökierrot osaavalla etäohjelmointi ohjelmistolla. CAM järjestelmää (Computer Aided Manufacturing – Tietokoneavusteinen valmistus) tarvitaan varsinkin törmäystarkasteluissa, koska suurnopeuskoneistuksessa tulee törmäystilanteessa paljon vahinkoa nopeuksien ollessa suuria ja karojen rakenteen ollessa herkempiä rikkoontumaan. Monissa CAM ohjelmistoissa on erikseen omat työkierrot ja parametrit suurnopeuskoneistukseen.

Peruseriaatteena on se että lastun paksuus ja syöttö pidetään vakiona huolimatta pinnanmuodon vaihteluista. Näistä asioista joudutaan kuitenkin välillä tinkimään esimerkiksi taskun koneistuksessa. Toimiva CAM järjestelmä mahdollistaa myös tehokkaan etäohjelmoinnin ja vain vähäiset ohjelmamuutokset ja asetukset koneella ennen tuotannon aloitusta.

#### **2.4.2. Jäysteenpoiston robotisointi**

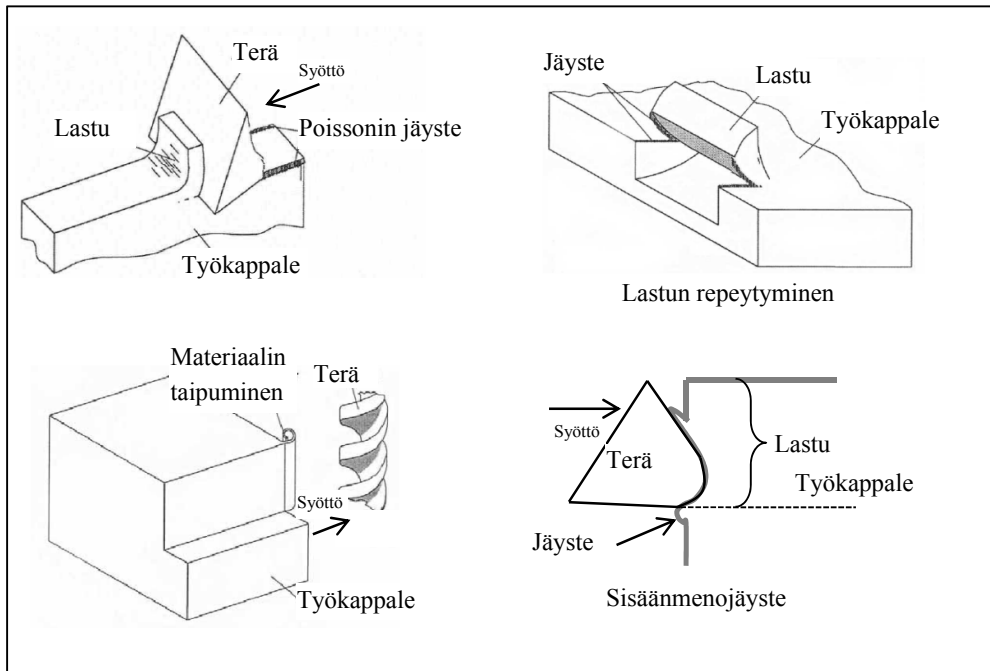
Materiaalinpoistolla robotiikassa tarkoitetaan robotilla tehtävää työstöä kuten valukappaleiden purseiden ja koneistuksessa syntyvien jäysteiden poistoa, sekä pinnan kiillotusta. Jäysteenpoisto on yksi potentiaalisimmista alueista tuotannon tehostamiseksi. Jäysteenpoisto suoritetaan yleensä käsityönä. Työ on raskasta, tapaturma-altista, likaista ja yksitoikkoista. Lisäksi prosessi ja käytettävät paineilmatyökalut ovat meluisia. Työn tehokkuus robotilla tehtynä voi nousta jopa 70 prosenttia, mikä on suurimpia tehokkuuden kasvuja robottisovelluksissa. [Glaser 2008]

Jäysteenpoistotehtävä voidaan katsoa suoritetuksi kun kappaleessa oleva jäyste tai purse on poistettu riittävästi, poistamatta kuitenkaan liiaksi perusmateriaalia ja pinnanlaadun säilyessä samanlaatuisena kappaleesta toiseen. Haasteelliseksi tehtävän tekee kappaleiden ja poistettavien elementtien mittavaihtelut koon ja paikoituksien suhteen. Vaihtelevissa tilanteissa ihminen on joustavampi, robotin etuna taas on tasalaatuisuus kappaleesta toiseen. [Otham 2007]

Jäysteet syntyvät lastuavissa työmenetelmissä eri tavoin. Kun lastu repeytyy irti työkappaleesta puhtaan leikkautumisen sijaan, työkappaleeseen jäävä osuus lastusta on

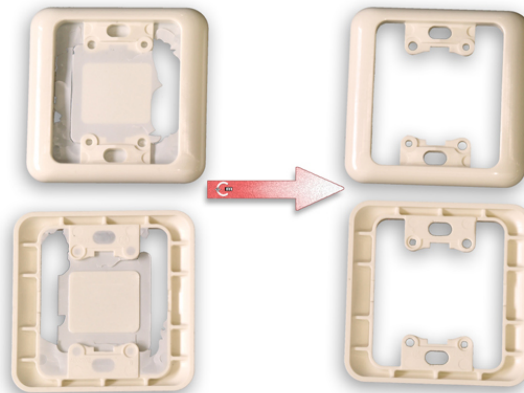
jäystettä. Työstävän terän tunkeutuessa työkappaleeseen kappaleen pinnalle syntyvää muodostumaa kutsutaan sisäänmenojäysteeksi.

Jäysteen synty perustuu yhteen tai useampaan muodonmuutosmekanismiin. Kuvassa 20 on erilaisia jäystetyyppejä. [Gillespie 1999]



**Kuva 20.** Jäysteen erilaisia muodostumistapoja koneistuksessa [Gillespie 1999]

Purseet taas syntyvät valukappaleissa, kun muotin jakotaso pääsee vuotamaan. Nämä ovat tyypillisiä kevytmetallivaluille ja kertamuovituotteille, kuva 21.



**Kuva 21.** Muovituote ennen ja jälkeen purseenpoiston [http://www.cmspa.com 2012]

Robotisoitu viimeistely [Bogue 2009]

- vähentää työntekijöihin kohdistuvia terveyshaittoja (raskas työ ja ympäristö)
- vähentää pullonkauloja muuten automatisoidussa tuotannossa
- vähentää työkalujen kulumista (tarkempi työkalun käyttö)
- parantaa läpimenoaikaa ja tuottavuutta
- parantaa laaduntuottokykyä ja tasalaatuisuutta
- vähentää viallisten tuotteiden syntyä.

Varsinaisen koneistuksen lisäksi jäysteenpoistoa suoritetaan myös CNC-koneilla, mutta robotit ovat ulottuvampia, halvempia ja joustavampia koneita kevyeen työstöön. Robottien heikkoutena on niiden epätarkkuus ja kevyempi rakenne. Robotit tuovat kuitenkin mahdollisuudet taloudellisesti tehokkaampaan tuotantoon. [Kim 2006]

Jäysteenpoiston robottisolut ovat usein varustettu jopa kymmenellä erilaisella työkalulla, mikä mahdollistaa useiden työvaiheiden suorittamisen yhdellä asetuksella. Robotti voi myös suorittaa kappaleenkäsittelyn varsinaisen prosessityön lisäksi samassa solussa. [Bogue 2009]

Robottityöstön toteutus valitaan kahdesta eri menettelytavasta. Robotti voi kuljettaa työkalua ja suorittaa jäysteenpoiston työkappaleen ollessa kiinni pitimessä. Toisessa tavassa robotti kuljettaa työkalualetta tarraimessaan ja työkalu on kiinnitetty robotin työalueelle. [Glaser 2008]

Jäysteenpoisto on hankalaa, koska poistettavan materiaalin määrä vaihtelee työkalusta toiseen, jopa saman kappaleen eri osissa. Tästä syystä tasaisen lastuvirran tuottaminen on vaikeaa ja työstössä syntyy vaikeasti hallittavia voimia. Tästä syystä työkalut tai työkalupitimet ovat yleensä joustavia, minkä avulla kompensoidaan työkalun mittavaihtelut (valukappaleet ja purseet), robotin tarkkuudesta johtuvat virheet ja työkalun kulumisen mittamuutokset. [Glaser 2008]

Joustavuutta saadaan myös joustavien nivelien, voimaohjauksen ja reaaliaikaisen radankorjauksen avulla. Avuksi on myös kehitetty erilaisia menetelmiä poikkeamien korjaamiseksi. Esimerkiksi menetelmä, jossa piezotoimilaite liikuttaa työkalualetta vä-

rinän kompensoimiseksi. Järjestelmä koostuu työkaluun kiinnitetystä kiihtyvyyssanturista, jonka avulla työkalun ja työkappaleen paikkapoikkeama mitataan. Tämä arvo syötetään piezotoimilaitteen ohjaukseen, jossa sumean logiikan avulla ohjataan työkappaleen liikkeitä. Tämän avulla työkalun ja työkappaleen välinen poikkeama pidetään mahdollisimman pienenä. [Moshen 2009]

Työkaluissa käytetään usein suuria karanopeuksia, mikä pienentää lastuamisvoimia ja sitä kautta vähentää robotin liikkeitä poikkeuttavia voimia. Robottien heikkoutena on lisäksi paikoitustarkkuus, mikä on tyypillisesti paljon huonompi kuin toistotarkkuus. Paikoitustarkkuudella tarkoitetaan robotin kykyä toteuttaa numeerisesti ohjelmoitu piste. Toistotarkkuudella tarkoitetaan robotin kykyä toistaa aikaisemmin opetettu paikoituspiste. Koska etäohjelmoinnissa paikoitukset ohjelmoidaan aina numeerisesti, tarkkuus jää toistotarkkuutta huonommaksi. Kuitenkin paikoitustarkkuuden virhe on pienempi kuin usein valukappaleissa olevat mittavaihtelut, eikä ole merkittävä ongelma jäysteenpoistosovelluksissa. [Black 2008]

## 2.5 Robotti solun osana

Robotteja käytetään tuotantoautomaatiossa tuotantosolujen kappaleenkäsittelyssä ja prosessoinnissa. Erilliset tuotantosolut voidaan liittää suuremmaksi joustavaksi valmistusjärjestelmäksi (FMS, Flexible Manufacturing System). Kuljettimet, työstökoneet ja robotit muodostavat laajan automaattisen järjestelmän, jonka ohjaus ja ohjelmointi on mahdollista hoitaa keskitetysti. Roboteilta vaaditaan hyvää liitettävyyttä muihin automaatiolaitteisiin, esim väylärajapinnat, I/O-liitännät, yms. Automatisoidut valmistusjärjestelmät mahdollistavat miehittämättömien tuotantojaksojen käytön.

Robottien käyttö osana laajempaa valmistusjärjestelmää edellyttää tarraimien rakenteiden ja apulaitteiden suurta joustavuutta. Kappaleenkäsittelyn joustavuus ratkaisee usein koko valmistusjärjestelmän kyvyn tuottaa pieniäkin sarjoja ja eriä tehokkaasti.

Robottien käyttö FM -järjestelmien rakennusmoduuleina antaa mahdollisuuden suurten ja nopeiden layout-muutosten tekemiseen. Standardoituja teollisuusrobotteja

voidaan käyttää työkalujen uudelleensuunnittelun jälkeen täysin uudenlaisissa tuotanto-tehtävissä. Työkalut ovat myös kehittyneet monipuolisiksi, joten aina ei edes työkaluja tarvitse uudelleensuunnittelua. [Kuivanen 1999]

## 2.6 Robottisolun suunnittelu

Robottisolun suunnittelu lähtee yleensä tuotannon tarpeista, ja hyvin usein solusta tehdään myös simulointimalli suunnittelun tueksi. Simuloinnista ja etäohjelmoinnista on enemmän kappaleessa 2.9 Etäohjelmointi ja simulointi.

Robotin valintaan vaikuttavat ensisijaisesti käyttötarkoitus, käsiteltävän tuotteen ominaisuudet, haluttu kapasiteetti, järjestelmän soveltuvuus tuotantoon sekä robotin hinta ja huollon saatavuus. Huomionarvoinen seikka on myös käytön sekä ohjelmoinnin helppous ja monipuolisuus.

Lisäksi on otettava huomioon [Kuivanen 1999]

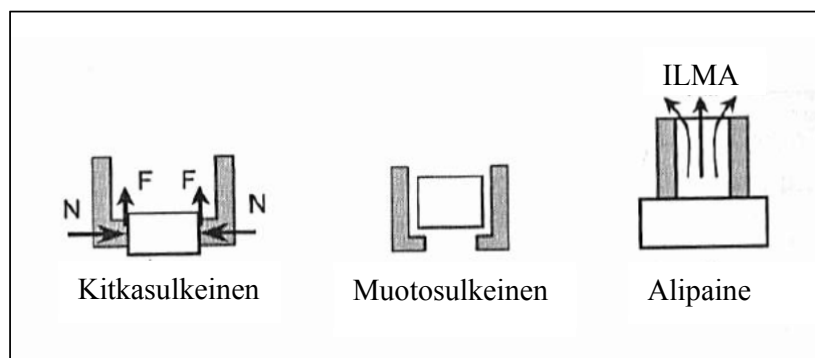
- todellinen käsittelykyky (tarrain + kuorma)
- ulottuvuus eri asennoissa
- tarvittavan tahtiajan saavuttaminen
- tarraimien ja työkalujen kiinnitys, sekä energian syöttö
- liitettävyyys oheislaitteisiin
- ohjelmointi
- käyttöönotettavat optiot.

Varsinaisen robotin kustannukset koko solusta on vain osa kokonaiskustannuksista. Varsinaisen robottikäsivarren lisäksi on huomioitava käyttöönoton ja koulutuksen synnyttämät kustannukset. Tärkeää on varautua myös robottisolun jatkuvaan kehittämiseen, näin solusta saadaan paras hyöty myös tulevaisuudessa.

## 2.7 Työkalut

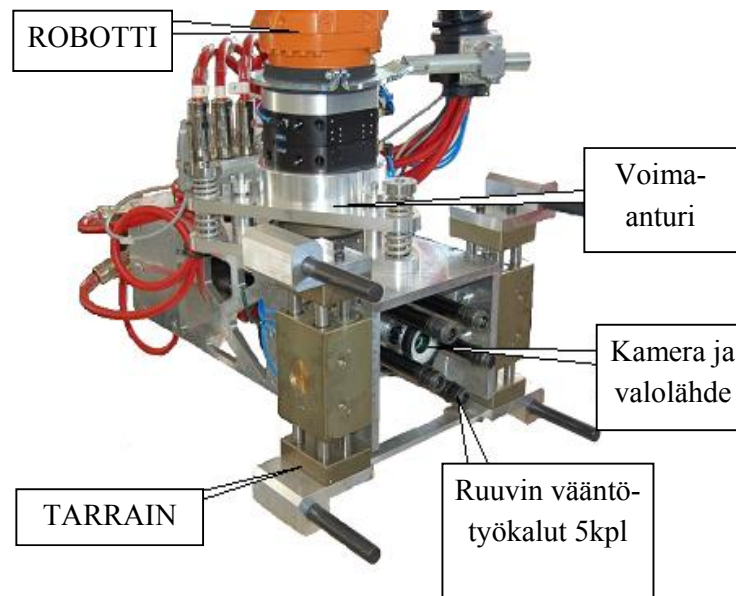
Työkalu on yksi tärkeimmistä robotin komponenteista, jonka avulla robotti tekee sille tarkoitettua tehtävää. Työkalut jaotellaan yleensä käyttöperiaatteen mukaan varsinaisiin työkaluihin ja tarraimiin. Työkaluilla toteutetaan valmistuksen prosesseja, kuten hitsaus, maalaus hionta jne. Tarraimilla toteutetaan kappaleenkäsittelytehtäviä.

Tarraimet voidaan jaotella monella eri tavalla. Yleensä jaottelu tehdään tarrainperiaatteen mukaan, jolloin puhutaan mekaanisista, alipaine- ja erikoistarraimista. Kappaleeseen voi tarttua usealla eri teknologiaperiaatteella. Käytetyimpiä tarrainperiaatteita ovat sormi- ja alipaine- ja magneettitarraimet. Tarrain saattaa sisältää useita eri tarrainmekanismeita, kuten alipainetekniikka ja lukitusmekanismi. Mekaanisia tarraimia käytetään tavanomaisten kappaleiden siirtämiseen, kun taas alipainetarraimet soveltuvat parhaiten levyäisten kappaleiden käsittelyyn. Kuvassa 22 on esitetty tarraintekniikoita. [Kuivanen 1999].



**Kuva 22.** Yleisimmät tartuntatavat kappaleenkäsittelyroboteilla. [Kuivanen 1999]

Työkaluihin liitetään jatkuvasti enemmän toimintoja ja älykkyyttä. Sähköiset vapaasti ohjelmoitavat sormitarraimet ja voimaohjauksen käyttö on lisääntynyt. Aikaisemmin erilaisilla joustavilla materiaaleilla ja mekanismeilla toteutettu jousto (passiivinen voimaohjaus) voidaan toteuttaa myös robotin älykkäällä joustolla eli ns. voimaohjauksella (aktiivinen voimaohjaus). Voimaohjaus perustuu voima-anturin käyttöön, joko työkalussa tai suoraan robotin ohjauksessa. Robotin liikeradat voidaan ohjelmoida siten että työkappaleeseen kohdistuu vakiovoima liikeradan aikana. Esimerkiksi tason hionnassa saadaan työkalun ja työkappaleen välinen voima pidettyä vakiona ja yksi tärkeä hiontaparametri vakioitua. Toinen tapa on käyttää ulkopuolista voimaa robotin liikkeen ohjaamiseksi, jolloin esimerkiksi asennustyössä robotti voi hakea pienintä voimaa liikerataansa ja saada kappale paikoilleen.



**Kuva 23.** Älykäs tarrain, autonrenkaan asennukseen. [The Institute of Robotics and Mechatronics 2010]

Kuvassa 23 on työkalu, jolla voidaan asentaa autonrenkas kuljettimella liikkuvaan autoon. Konenäköä hyödynnetään paikallistamaan vanne ja vanteen paikka autossa. Voimaohjausta hyödynnetään pitämään asennuksen aikana robotti oikeassa paikassa suhteessa liikkuvaan kohteeseen. Älykkäät työkalut yhdistettynä voimaohjaukseen antaa paljon uusia mahdollisuuksia robotiikan hyödyntämisessä. [The Institute of Robotics and Mechatronics 2010]

### 2.7.1. Jäysteenpoiston työkalut

Jäysteenpoistorobotin työkalun muodostavat karaa pyörittävä moottori, jäystetyökalu ja työkalunpidin, kuva 24. Työkalu voidaan asentaa robotin työkalulaippaan tai kiinteästi robottisoluun, tällöin robotti käsittelee työkalualetta.

Karamoottori pyörittää työkalua tai liikuttaa sitä aksiaalisesti edestakaisin (viilat). Moottorit on jaettu seuraavasti pyörimisnopeuden mukaan [Pennock 2006]:

- Suurnopeuskarat yli 15 000 kierrosta/min pyöriville viiloille ts.koneviilat.
- Keskinopeat karat (10 000...15 000 kierrosta/min) pyöriville harjoille alle 60 mm halkaisijoille.
- Hitaasti pyörivät työkalut (500...1500 kierrosta/min).

Paineilmamootorit ovat yleisimpiä, koska ovat edullisia, kestäviä ja suhteellisen kevyitä robotin käsiteltäviksi. Jos vaaditaan korkeampia kierrosnopeuksia (yli 30 000 kierrosta/min) tai kierrosnopeuden säädettävyyttä niin valitaan sähkötoiminen kara. Sähkötoimisien karojen käyttö on lisääntynyt myös paremman energiahyötysuhteen ansiosta. [Pennock 2006]



**Kuva 24.** Jäysteenpoiston työkaluja. [<http://www.ati-ia.com/products> 2011]

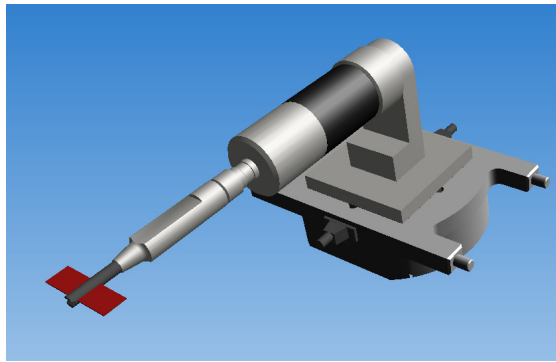
Jäystetyökalut voidaan jaotella joustaviin työkaluihin, pyöriviin koneviiloihin ja aksiaalisesti edestakaisin liikkuviin viiloihin. Joustavilla työkaluilla tässä yhteydessä tarkoitetaan työkalun rakenteellista joustoa kuten nauhahiomakoneita, harjoja ja hiomalaikkoja. Harjat ovat alumiinioksidilla tai silikoni-karbidilla käsiteltyä polyamidia tai terästä, kuva 25.



**Kuva 25.** Nokka-akselin jäysteenpoistoa harjatyökalulla. [Weilercorp 2012]

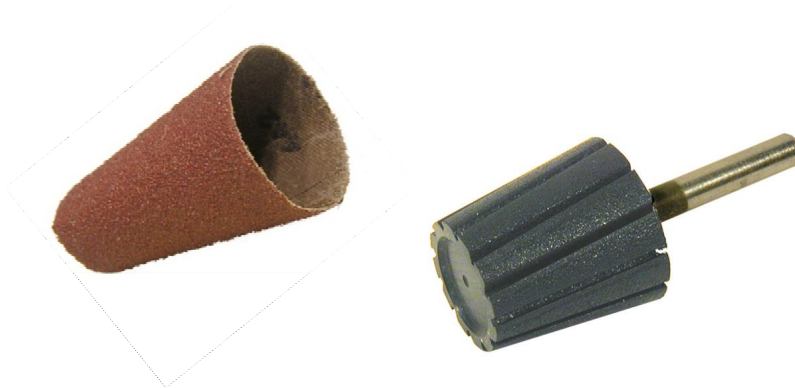


Hiomanauhat ja kankaat valmistetaan liimaamalla hiomaraheet synteettisistä kuiduista valmistettuun kankaaseen. Hiomaraheet ovat mineraaleja, kuten piikarbidia tai alumiinioksidia ja kuidut polyamidia (nylon). Hiomakankaaseen perustuvia jäysteenpoistotyökaluja ovat myös Flap- työkalu ja kartiohiomapaperit. Flap- työkalussa on suorakaiteen muotoinen hiomakangas kiinnitetty pyörivään akseliin. Kuvassa 26 on Flap- työkalu. Joustava hiomakangas irrottaa jäysteen tehokkaasti erikokoisista ja muotoisista taskuista ja onkaloista. Flap- työkalulla on useita nimityksiä, kuten läpsytin, räpaskä ja fläppi.



***Kuva 26.** Flap- työkalu.*

Hiomakartio koostuu kartion tai lieriön muotoon valmistetusta hiomanauhasta ja työkalunpitimeen kiinnitettävästä kumikarasta, kuva 27. Kuminen kara joustaa ja kompensoi paikoitusvirheitä. Hiomanauhojen käyttöä hankaloittavat nauhanvaihdon ongelmat. Nauha joudutaan vaihtamaan useammin kuin metalliset työkalut, joten vaihdon automatisointi tuo lisää työvaiheita ja kompleksisuutta tuotantosoluun. Hiomanauhasovellukset ovat kuitenkin yleinen menetelmä robotisoidussa jäysteenpoistossa.



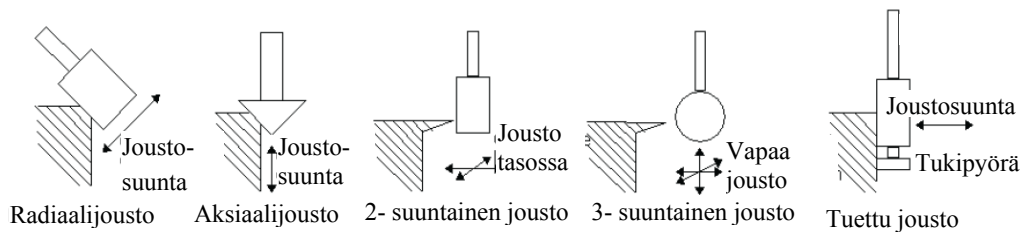
**Kuva 27.** Hiomakartio ja kumikara. [www.a-palojoki.fi 2012]

Yleisimpiä jäystetyökaluja ovat pyörivät viilat ns. koneviilat, kuva 28. Koneviiloja on saatavilla eri materiaaleista, yleensä kovametallia ja erilaisilla terägeometrioilla. Työkalun valintaan vaikuttavat terän geometria, materiaali, jäystettävä materiaali ja jäysteen muoto sekä koko. Koneviiloissa käytetään suuria pyörimisnopeuksia ja yleensä myötäjärsintää. Myötäjärsinnällä minimoidaan työkalun värähtelyjä. [Pennock 2006]



**Kuva 28.** Erilaisia koneviilageometrioita. [www.molyteam.net 2012]

Joustoa käytetään hyväksi kompensoimaan jäysteenpoistoympäristön mittavaihteluita, joita on käsitelty sivulla 33. Joustoa voi syntyä työkalusta tai työkalunpitimeen rakennetusta joustoyksiköstä. Jousto voi olla aksiaalista, radiaalista tai molempia ja jopa joka suuntaan joustavia työkaluja. Kuva 29. Jäysteenpoisto- ja viimeistelykarat voidaan varustaa myös anturoinnilla. Asentoa ja kierrosnopeutta seuraamalla voidaan tehdä esimerkiksi radankorjauksia työvaiheen aikana. [Pennock 2006]



**Kuva 29.** Jäysteenpoiston työkalujen joustosuuntia. [Pennock 2006]

Säteen suuntaan joustavat työkalut ovat parhaimmillaan työkappaleen jakotason purseiden poistossa. Kara voidaan asentaa robotin työkalulaippaan tai kiinteästi jäysteenpoistosoluun. [http://www.ati-ia.com/products 2011]



**Kuva 30.** Aksiaalinen joustokara jäysteenpoistoon. [<http://www.ati-ia.com> 2011]

Kuvassa 30 on aksiaalinen joustokara jäysteenpoistoon. Työkaluna käytetään 45 asteen kulmassa olevaa teräsärmää. Tämä geometria yhdistettynä aksiaaliseen joustoon joustaa sekä sivu-, että pystysuuntaan. Sopiva jousto pitää lastuamisvoiman vakiona, mikä on yleensä hankala hallita robotisoidussa jäysteenpoistossa. Aksiaalinen jousto kompensoi ratapoikkeamia, sekä kappaleen muotovaihteluja. Työkalu sopii työkappaleen reunajäysteiden poistoon. [<http://www.ati-ia.com/products> 2011]



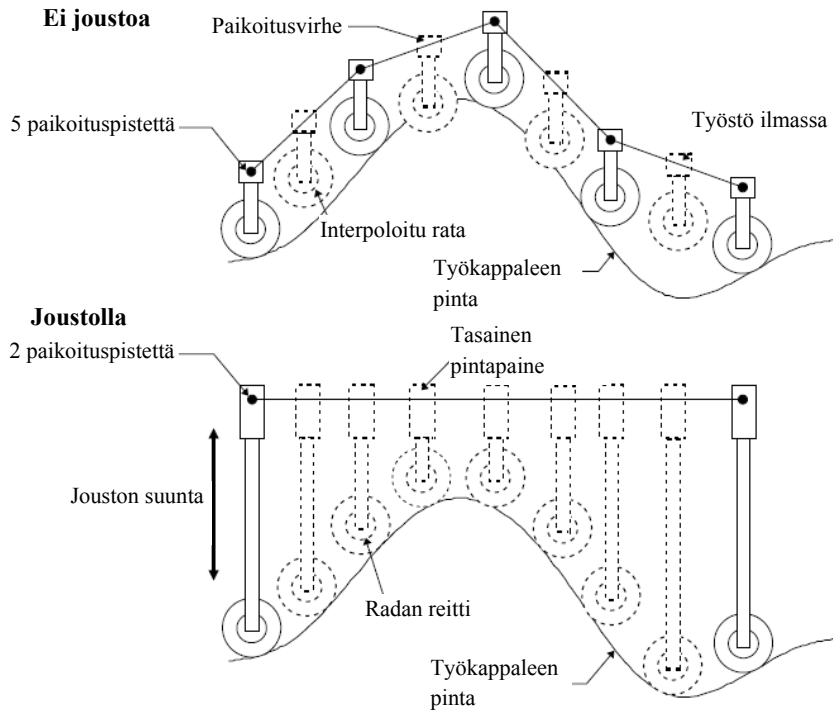
**Kuva 31.** Aksiaalinen joustokara harjatyökalulla. [<http://www.ati-ia.com> 2011]

Viimeistelyhiontaan sopiva työkalu on esimerkiksi kuvan 31 hidaskierroksinen suurimomenttinen paineilmatyökalu aksiaalijoustolla. Harjatyökaluilla poistetaan pieniä jäysteitä hyvin monimuotoisilta muodoilta. [<http://www.ati-ia.com/products> 2011]

### 2.7.2. Voimaohjaus jäysteenpoistossa

Voimaohjauksen käyttö jäysteenpoistosovelluksissa tuo paljon uusia mahdollisuuksia prosessin parempaan ohjattavuuteen. Kuten aiemmin on todettu, robotilla tehtävän jäys-

teenpoiston apuna käytetään hyvin usein joustoa kompensoimaan paikoitusvirheitä. Jousto voi olla työkalussa, työkalunpitimessä tai robotin ohjauksessa. Voimaohjauksen avulla saadaan pidettyä vakiovoima työkappaleen ja työkalun välillä. Voimaohjaus voidaan käsittää myös joustoelementtinä, koska robotti voidaan ohjelmoida myötäilemään kappaleen piirteitä. Ohjelmoinnissa riittää myös pienempi määrä paikoituspisteitä, koska järjestelmä seuraa voiman avulla pinnan muotoa.



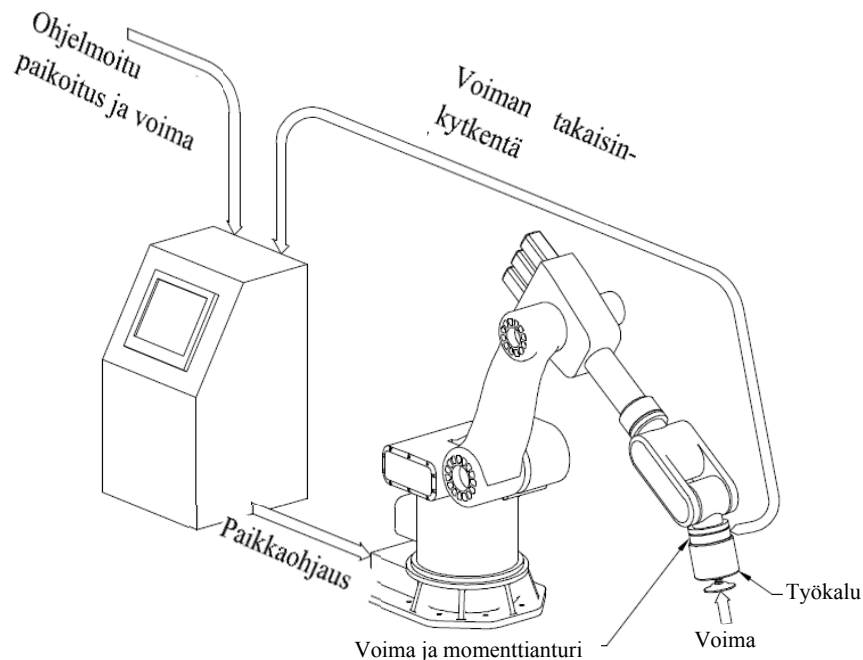
**Kuva 32.** Jouston periaate pinnan viimeistelyssä. [PushCorp 2011]

Kuvassa 32 on esitetty joustavan menetelmän etu jäykän työkalun käyttöön. Joustavan työkalun avulla pinta tulee viimeistelyä kahdella paikoituspisteellä, voimaohjaus huolehtii paikoituksesta. Ilman voimaohjausta tarvitaan huomattavan paljon paikoituspisteitä. Suuri paikoituspisteiden määrä ei ole ongelma etäohjelmointiympäristössä, mutta sekään ei takaa hyvää lopputulosta, koska kappaleiden dimensiot ja jäysteen koko saattaa vaihdella. Voimaohjauksen avulla saadaan yksi tärkeä prosessiparametri, työkalun paine työkappaletta vasten, helposti hallintaan.

**Passiivinen voimaohjaus** tarkoittaa työkalunpitimessä olevaa ohjauksella muutettavaa joustovoimaa, mutta voimasta ei ole takaisinkytkentää. Voima vaihtelee olosuhteiden muuttuessa.

**Aktiivinen voimaohjaus** on järjestelmä, missä voima mitataan ja pidetään halutun suuruisena ympäristöolosuhteista riippumatta. Aktiivinen järjestelmä voi olla robotin ohjaukseen integroitu tai ulkoinen järjestelmä.

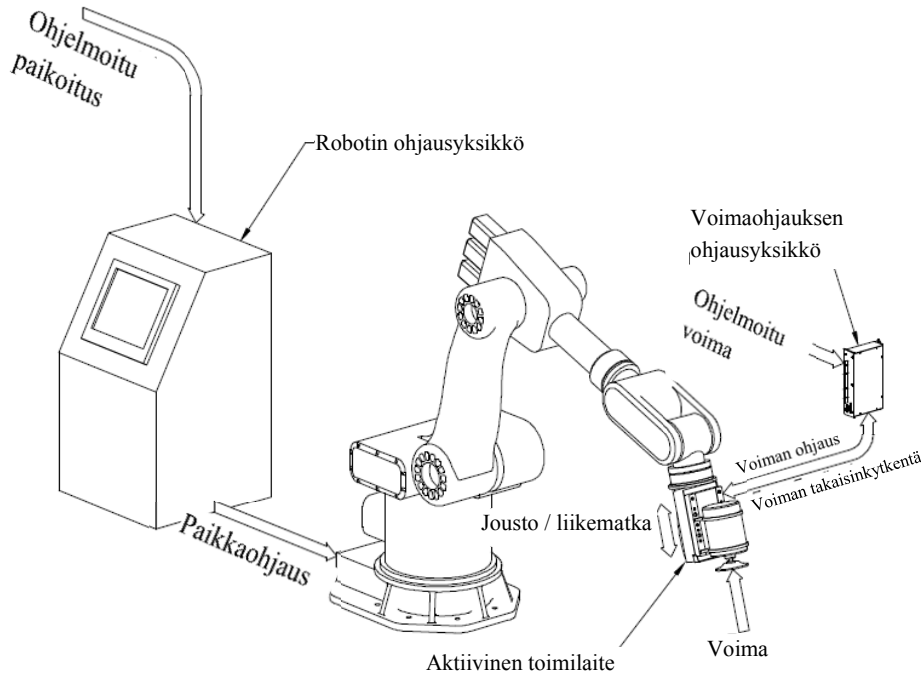
Integroitu voimaohjaus sisältää anturin työkalulaipan ja työkalupitimen välissä. Anturi välittää voiman ja momentin arvot ohjaukselle työkalukoordinaatiston jokaisen vapausasteen suuntaan. Järjestelmä on merkkiriippuvainen, jokaisella robottivalmistajalla on oma järjestelmänsä. Integroidun järjestelmän avulla saadaan joustosuunta valittua vapaasti. Ohjausta voidaan käyttää ohjelmoinnin apuna liikuttamalla robottia työkaluun vaikuttamalla. Voimaohjauksen nopeus määräytyy robotin kyvyistä toteuttaa nopeat liikkeet. Kuvassa 33 on integroidun järjestelmän pääosat. [PushCorp 2011]



**Kuva 33.** Ohjaukseen integroitu voimaohjaus. [PushCorp 2011]

Ulkoinen voimaohjaus kontrolloi voimaa yleensä yhden vapausasteen suuntaan. Kuvassa 34 ulkoinen voimaohjaus kontrolloi työkalun Z-akselin suuntaista voimaa. Ul-

koinen yksikkö on nopeampi kuin integroitu, koska liikeohjaus tapahtuu erillisessä yksikössä eikä kuormita robotin ohjausjärjestelmää. Se on merkkiriippumaton järjestelmä ja voidaan lisätä myös käytössä oleviin robotteihin.



*Kuva 34. Ohjaukseen liitetty voimaohjaus. [PushCorp 2011]*

Voimaohjaus on toistaiseksi paljon kalliimpi vaihtoehto passiivisille joustoelementeille, mutta oikein valittuna antaa paljon joustavuutta ja varmuutta jäysteenpoistosoveluksiin.

### 2.7.3. Jäysteenpoistosolun suunnitteluprosessi

Robotisointien ja automatisointien asettamat perusvaatimukset, kuten vaadittava kapasiteetti, käsiteltävät prosessit ja työkappaleiden kuljetukset ovat myös jäysteenpoistosolun suunnittelun perusvaatimuksia jotka on ratkaistava. Jäysteenpoiston suunnittelussa täytyy huomioida muutkin jäysteenpoistomenetelmät, eikä robotisoitu jäysteenpoisto ole aina paras vaihtoehto. Työn aiheena on kuitenkin robotisoidut ratkaisut, joten muut jäysteenpoistomenetelmät jätetään käsittelemättä.

Jäysteenpoiston robotisoinnissa erityisiä tekijöitä ovat menetelmät lastuamisvoimien pienentämiseksi, eri syistä johtuvien paikoitusvirheiden eliminointi ja etäohjelmointi. Paikoitusvirheitä syntyy useista paikoista, kuten aikaisemmin on esitetty. Myös työkalupaleen suunnitteluvaiheessa voidaan kontrolloida jäysteen syntyä. Työkalun valinnassa tulee huomioida työkalun sopivuus prosessiin, kestävyys, automatisoitu teräriikon havainnointi, kulumisen kompensointi ja työkalunvaihtoprosessi.

Sovelluksessa voidaan käyttää hyödyksi suurnopeuskoneistuksen ja CAM- järjestelmiin kehitettyjä menetelmiä. Etäohjelmointimenetelmät helpottavat lastunpaksuuden pitämistä vakiona kaarevilla pinnoilla. Etäohjelmointi yhdistettynä sopivaan teknologiaan, esimerkiksi joustaviin työkaluihin tai voimaohjaukseen, tuottaa kilpailukykyisen robottisovelluksen. Joustavat ohjausjärjestelmät mahdollistavat myös prosessiparametrien muuttamisen työvaiheen aikana. Esimerkiksi materiaalia poistavan työkalun kulumisen voidaan kompensoida lisäämällä voimaa tai hidastamalla syöttönopeutta tasaisen laadun saavuttamiseksi. Työkalu voidaan myös mitata ja määrittää uudelleen säännöllisesti. [Bogue 2009]

Jäysteenpoiston robotisointi voidaan toteuttaa eri tavoin ja voidaan päästä hyvään lopputulokseen. Ratkaisuissa on eroja esimerkiksi laitteiden hinnoissa, tehokkuudessa ja muunneltavuudessa. Ratkaisun valinta ei ole yleensä yksikäsitteinen eikä helppo. [Tolinski 2006]

Jäysteenpoistomenetelmän valintaan vaikuttavia tekijöitä ovat [Tolinski 2006]

- jäystetyyppi
- jäysteen koko ja mittavaihtelut
- työkalupaleen materiaali
- työkalun materiaali ja geometria
- työkalun rakenteellinen jousto (varren jousto, hiomalaikan jousto)
- työkalun toiminnallinen jousto (pitimeen rakennettu joustomekanismi).

Suunnilleen samanlaisen listan on Pennock laatinut jäystetyökalun valintaa käsitellessään [Pennock 2006]:

- Työkalun geometrian sopivuus määrätyn materiaalin poistossa on tarkastettava.

- Työkalujouaston valinta jäysteen koon ja paikan mukaan, kuten passiivinen jousto, voimaohjauksen käyttö.
- Työkalun lastuamisarvojen selvitys, kuten lastuamisvoimat, nopeus, kierrosnopeus työkappaleen materiaalille.
- Karan käyttövoiman valinta, paineilma- vai sähkötoiminen.

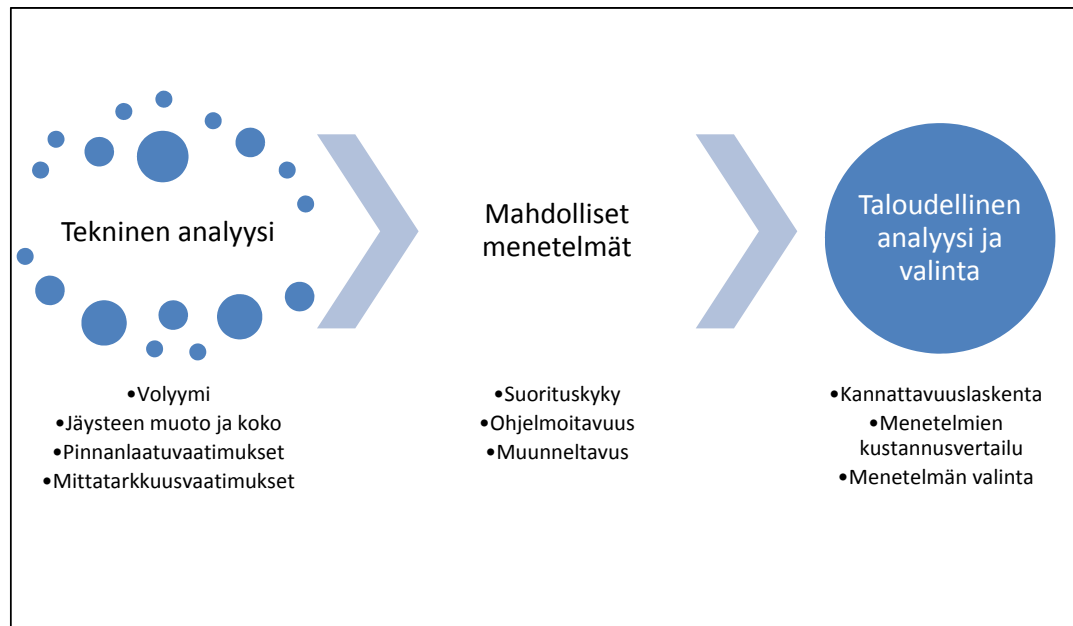
Kolmas lähestymistapa jäysteenpoiston suunnitteluun löytyy PushCorp Inc. julkaisusta voimaohjattujen järjestelmien ohjelmoinnista. [PushCorp Inc. 2012]

- Jäysteenpoistomenetelmän valinta, kappale vai työkalu robotin käsivarteen.
- Ohjelmointityökalun valinta, opettamalla vai etäohjelmoinnin avulla ohjelmointi.
- Tämän jälkeen valinta passiivisen ja aktiivisen voimaohjauksen väliltä.

Tärkeä parametri jäysteenpoistossa ja viimeistelyssä on työkalun asento suhteessa työkappaleeseen. Tutkimuksissa on havaittu että jopa kahden asteen muutos työkalun kulmassa vaikuttaa merkittävästi lopputulokseen. Konseptin testausvaiheessa etäohjelmointiympäristön käyttö helpottaa oikeiden kulmien löytymistä. Etäohjelmoinnin avulla saadaan kulmat tarkasti haluttuun arvoon, sekä voidaan muokata kokonaisia paikoituspisteryhmiä kerralla. [PushCorp Inc. 2012]

Jäysteen määrittely on usein vaikeaa, koska sen mittaaminen on hankalaa. Tärkeitä parametreja ovat jäysteen koko, materiaali, sijainti ja jäysteen tila (esim. työstökarjennut koneistusjäyste). Usein huomio kiinnittyy vaikeimman jäysteen poistoon ja menetelmä valitaan sen mukaan. Robotit voidaan varustaa tarpeellisella määrällä työkaluja, eikä erilaiset jäysteet ole silloin ongelma. Lastuavaan työstöön perustuvat jäysteenpoiston menetelmät poistavat usein myös materiaalia varsinaisesta kappaleesta. Tällöin on huolehdittava, etteivät kappaleen mitta- ja muototoleranssit ylity. [Tolinski 2006]





**Kuva 35.** Jäysteenpoistomenetelmän valintaprosessi. [Tolinski 2006]

Menetelmän valinta voidaan kuvata kuvan 35 mukaisena prosessina. Menetelmien valintaprosessi voi olla joko yrityksen ja erehdyksen kautta kokeilemista tai systemaattiseen päätöksentekoon perustuvaa. Koska prosessi on monimutkainen, tulisi kaikki prosessitekijät kuten volyymi, dimensiot, kustannukset ym. määritellä. Tämän jälkeen tarkastelu tulisi tehdä systemaattisesti useamman jäysteenpoistomenetelmän kannalta. Lisäksi tulisi kiinnittää huomiota jäysteen muodostumiseen, voiko sitä muuttaa helpommin poistettavaksi, koon, paikan ja tyypin mukaan. Paras vaihtoehto olisi kehittää tuotantomenetelmää siten että jäysteenpoisto voitaisiin välttää kokonaan. [Tolinski 2006]

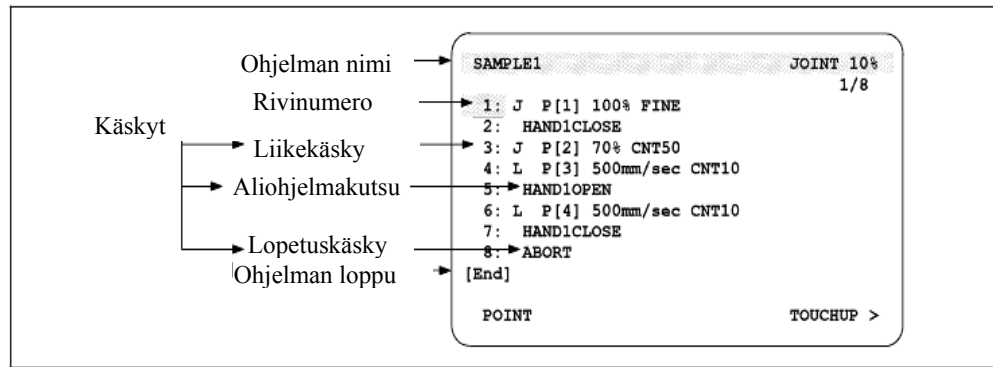
## 2.8 Teollisuusrobotin ohjelmointi

Robottien ohjelmointi voidaan jakaa online- ohjelmointiin ja etäohjelmointiin eli offline- ohjelmointiin. Yleensä robotteja ohjelmoidaan online, jolloin robotti on poissa tuotantokäytöstä ja ohjelmointi suoritetaan käsiohjaimen avulla. Etäohjelmointi suoritetaan erillisellä tietokoneohjelmistolla ja valmis ohjelma siirretään robotin ohjausjärjestelmään. Etäohjelmointi voi olla tekstipohjaista tai mallipohjaista. Tekstipohjaisessa ohjelmoinnissa ohjelmointi tapahtuu kirjoittamalla robottiohjelma tekstinä robotin ohjelmakielellä. Mallipohjaisessa ohjelmoinnissa käytetään hyväksi 3D- malleja, joiden piirteitä käytetään ohjelmaratojen luonnissa. Robotin ohjelmaan voidaan ohjelmoida liikekäsikyjä, erilaisia ehto- ja silmukkarakenteita, ulkoisien laitteiden ohjauskäsikyjä ja paikoituksiin liittyvää dataa. [Graig 2005]

### 2.8.1. Fanuc- teollisuusrobotin ohjelmointi

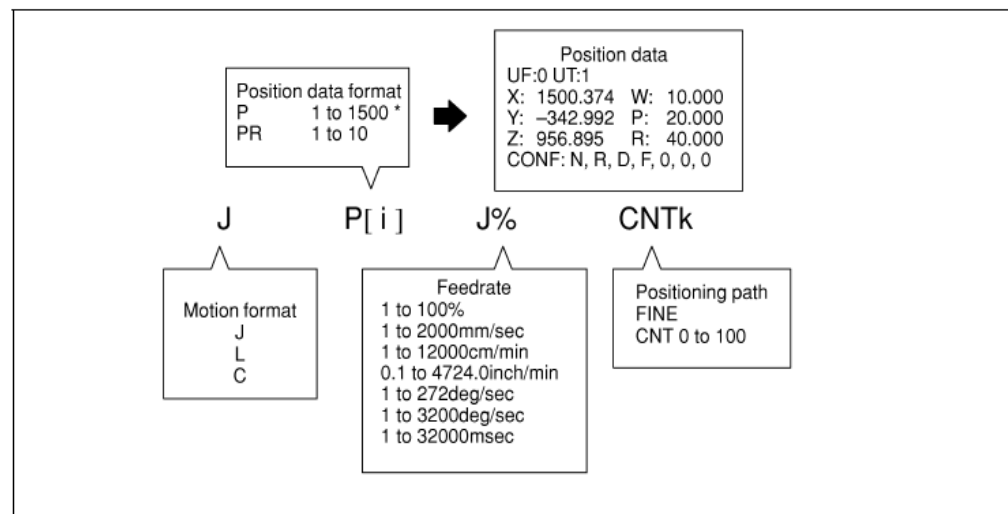
Fanuc- robotteja voidaan ohjelmoida käsiohjaimen tai erillisen tietokoneen avulla. Ohjelmointikielinä Karel ja TP- ohjelmointikielet (Teach Pendant). Karel- kielinen ohjelma on korkeamman tason ohjelmointikieli, jonka avulla voidaan toteuttaa monimutkaisia funktioita ja algoritmeja. Karel- kielinen ohjelma ohjelmoidaan tietokoneella omalla ohjelmistolla. TP- kieliset ohjelmat ovat käsiohjaimella tai etäohjelmointiympäristössä ohjelmoitavia ohjelmia. Se on yleisimmin käytössä oleva kieli. Käsiohjaimen avulla voidaan tehdä ja muokata TP-ohjelmia. TP- ohjelmat voidaan tallentaa tekstimuodossa (ASCII) \*.LS ja \*.PE tai konekielisenä (Binäärimuotoisena) \*.TP. Vanhemmissa Fanuc- ohjauksissa tekstimuotoinen tiedosto täytyy muuttaa konekieliseen muotoon robotille siirrettäessä.

Ohjelma koostuu liikekäsikyistä, I/O- komennoista ,rekisterien käsittelystä ja ohjelman kulkuun liittyvistä komennoista. Jokainen käsky on ohjelmassa omalla rivillään ja rivit ovat numeroitu. Järjestelmän tarkoituksena on suorittaa näitä komentorivejä rivi kerrallaan, kunnes ohjelman lopetusmerkki saavutetaan, kuva 36.



**Kuva 36.** Fanuc- robotin ohjelmarakenne. [Fanuc 2001]

Kaikki robotin paikoitukset ovat tallennettu joko suorakulmaisiin koordinaatein (xyz paikoitusarvot) ja orientaatiot kulma-arvoina (w,p ja r kulmat) tai nivelien kulma-arvoina. Robotin tehtävä on siirtää TKP haluttuun paikkaan (x,y,z) ja orientaatioon (w,p,r). Kuvassa 37 on liikekäskyn elementit ja vaihtoehdot.

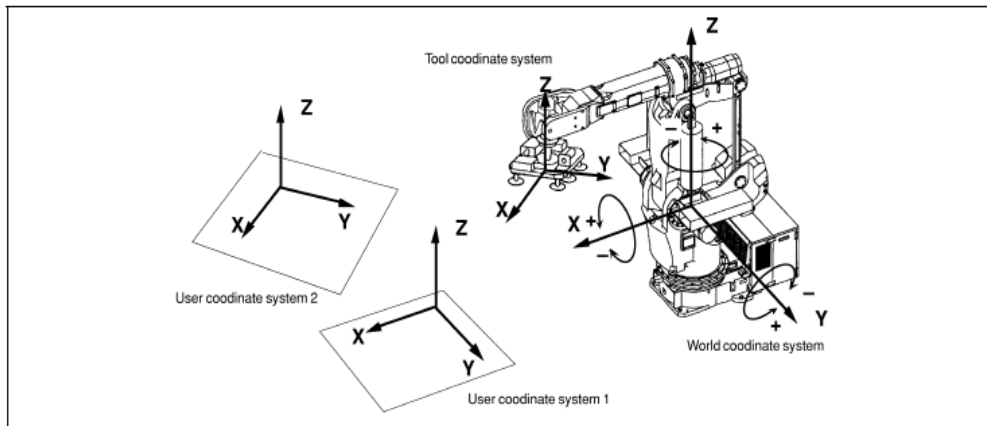


**Kuva 37.** Fanuc- robotin liikekäskyn sisältö. [Fanuc 2001]

Liikekäsky pitää sisällään liiketavan valittuun paikoituspisteeseen, nopeuden ja paikoitustarkkuuden. (Käyttöohje: liiketavan ohjelmointi Liite 2 s.41)

Robotti laskee jatkuvasti TKP:n paikkaa työavaruudessaan ja vertaa sitä tavoitepisteeseen koordinaatteihin. Robotti käyttää apuna koordinaattijärjestelmiä nivelien ohjauksessa. Maailmankoordinaatisto (Worldframe) on kiinteä peruskoordinaatisto, jonka kautta laskeminen tapahtuu. Toinen kiinteä koordinaatisto on työkalulaipan koordinaatisto (ToolFrame 0). Kuva 38. Esimerkiksi ohjelmoimme Tool0:n origolle (TKP) jonkin

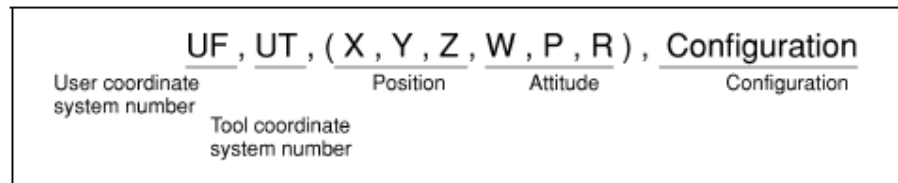
paikan työalueella (maailmankoordinaatistossa). Tämän jälkeen robotin ohjain laskee ja toteuttaa sellaiset nivelkulma-arvot eri akseleille, jotta robotin oikea asento saavutetaan. Tai toisinpäin, voimme ajaa robottia käsiä jolla haluttuun paikkaan ja orientaatioon. Silloin paikoituspisteen arvot tallennetaan. Tämän jälkeen robotti voi palata samaan paikoituspisteeseen tarpeen vaatiessa.



**Kuva 38.** Fanuc- robotin koordinaatistot. [Fanuc 2001]

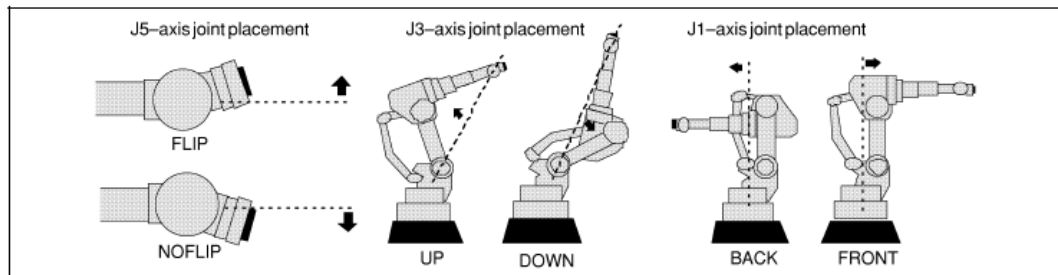
Kun robotin työkalu vaihdetaan fyysisesti, täytyy ohjausjärjestelmälle kertoa uuden työkalun työkalupisteen paikka ja asento suhteessa Tool0:n koordinaatistoon. Tätä varten määrittelemme robotille uuden työkalun koordinaatiston (TFrame 1-10). Määrittely voidaan tehdä numeerisesti tai käyttämällä robottia mittavälineenä. (Käyttöohje: työkalun määrittely). Samoin voidaan opettaa työalueelle käyttäjän koordinaatistoja (UFrame 1-10), mitkä helpottavat paikoitusten numeerista käsittelyä ja työkalun liikuttamista haluttuihin suuntiin. Ainoastaan yksi työkalu ja koordinaatisto voi olla aktiivisena. Aktiivinen koordinaatisto ja työkalu tulee aina opetettavan paikoituksen sisällöksi.

Paikoituspisteet on pystyttävä määrittelemään yksikäsitteisesti, ts. ohjaimen on pystyttävä valitsemaan useista TKP:n paikoituksen toteuttavista asennoista oikea. Kuvassa 39 on paikoituksen määrittelyyn liittyvät muuttujat.



**Kuva 39.** Fanuc- robotin paikoituksen määrittely. [Fanuc 2001]

Paikoituksen määrittely koostuu käytettävästä koordinaatistosta (UF) ja työkalusta (UT), TKP:n paikka ja orientaatiotiedosta, sekä robotin konfiguraatiosta. Konfiguraatio-tieto koostuu nivelvarsien keskinäisistä asennoista, kuva 40 ja nivelien 1,4 ja 6 kierros-luvusta, ts. ilmaistaan millä kierroksella yli ko. akseli on. ( $1=180^\circ-539^\circ$ ,  $0=(-179^\circ)-179^\circ$ ,  $(-1) = (-539^\circ)-180^\circ$ .



**Kuva 40.** Fanuc- robotin asentokonfiguraatiot. [Fanuc 2001]

Ohjausjärjestelmä käsittelee paikoituspisteitä koordinaatistoina, kuten työkalu- ja käyttäjän koordinaatistojakin. Paikoitus on toteutettu silloin kun valittu työkalun koordinaatisto on samassa paikassa ja asennossa liikekäskyn paikoituspisteen kanssa. Esimerkiksi Kuvan 41 pisteen P[1] määrittelykset tarkoittavat seuraavaa:

- Userframe, UF=0, piste on määritelty robotin peruskoordinaatistossa.
- Usertool, UT=7, työkalukoordinaatisto 7 käytössä.
- Koordinaatit ja kiertokulmat ilmaisevat työkalu 7:n paikan ja asennon robotin peruskoordinaatistossa.
- CONF: NUT = Konfiguraatio Noflip, Up, front, 000= 1, 4 ja 6 akselit ensimmäisellä kierroksellaan. Nämä kiertoakselit kiertyvät yli yhden kierroksen, saavat siis saman asennon jokaisella kierroksella.

Position Detail				JOINT 15 %	
P[1]	UF:0 UT:7	CONF:NUT 000			
X	2289.219	mm	W	-150.164	deg
Y	100.000	mm	P	-28.900	deg
Z	1173.398	mm	R	-35.989	deg
MIKA					
				PAUSED	1/4
1:J	@P[1]	100% FINE			
2:J	P[2]	5% FINE			
3:L	P[3]	500mm/sec CNT50			

**Kuva 41.** Fanuc- robotin paikoituksen määrittelydata. [Fanuc 2001]

Ohjelmoinnin peruskaava käsiohjaimella opetettaessa on seuraava:

- Käytettävän työkalun ja koordinaatiston valinta aktiiviseksi.
- TKP:n siirto haluttuun paikkaan ja liikekäslyn määrittäminen ko. paikoitukseen.
- TKP:n siirto seuraavaan paikkaan ja liikekäslyn määrittäminen.
- Työvaiheita toistetaan kunnes koko työkierto on valmis.
- Lopuksi suoritetaan ohjelman testaus.

Ohjelmat voidaan tehdä myös etäohjelmointiympäristössä. Etäohjelmointiympäristö kalibroidaan vastaamaan reaalirobottisolua. Virtuaalimallissa käytetään samoja työkalujen ja koordinaatistojen määrittelydataa kuin oikealla robotilla. Ne voidaan määrittää joko mallissa tai oikealla robotilla ja siirtää ne sitten sieltä toiseen ympäristöön. Minissä robotille viedään vain liikekäskyt ja paikoitustiedot virtuaalimaailmasta. Usein viedään myös työkaluvaihtokutsut, I/O-komentoja työkalun ja oheislaitteiden ohjaamiseksi. Lisäksi kommentointi, mikä onkin huomattavasti helpompaa tietokoneella kuin käsiohjaimella tehtynä. Käyttäjän koordinaatisto kiinnitetään työkappaleeseen ja paikoitukset muodostetaan siihen. Etäohjelmointiympäristössä on helppo muokata suuriakin määriä paikoituksia nopeasti samanaikaisesti.

## 2.9 Etäohjelmointi ja simulointi

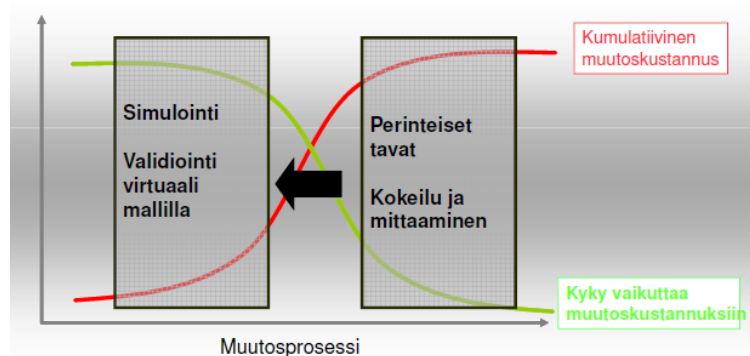
Tietokonesimulointi tarkoittaa reaali maailman mallintamista tietokoneelle ja mallin toiminnallisuuden testausta esim. eri parametrein. Se voi olla graafista tai numeerista. Graafinen simulointi on lisääntynyt tietokoneiden ja ohjelmistojen kehittymisen myötä. Graafinen simulointi tuottaa myös visuaalisen kuvan simulointikohteesta, mikä on tärkeää varsinkin robottisimuloinneissa.

”Simulointi on kokeilua yksinkertaistetulla (tietokoneella tehdyllä) dynaamisella systeemillä, jotta voisimme ymmärtää oikeaa systeemiä paremmin ja kehittää sitä paremmaksi.” [Robinson 2004]

Nykyaikainen solusuunnittelu lähtee simulointimallin teosta. Tuotantolinjasta tai sen osasta mallinnetaan 3D-malli. Ostosista löytyy yleensä valmis malli siirrettäväksi simulaatioon. Yleisesti käytettyjä tiedostomuotoja ovat: STL, IGES, STEP ja SAT. Mallilla voidaan

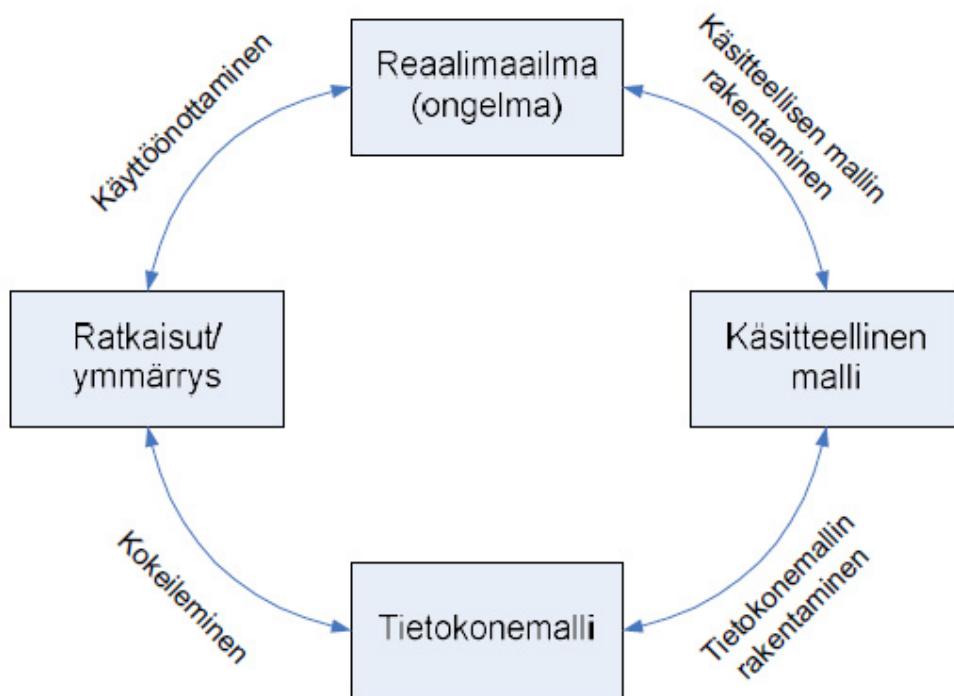
- kokeilla erilaisia layout-vaihtoehtoja ja ulottuvuustarkastelua
- määrittää työsekvenssit ja arvioida tahtiaikoja
- tehdä robotin, työkalun, kiinnittimien ja oheislaitteiden törmäystarkastelu
- testata työkaluvaihtoja sekä työkalun ulottuvuus- ja luoksepääsy tarkastelu.

Simuloinnissa voidaan kokeilla laitteiston toimivuutta ennen kuin kalliita komponentteja on hankittu. Esimerkiksi robotin ulottuvuustarkastelun avulla nähdään onko valittu oikea robottityyppi. Kuvassa 42 näkyy kuinka simuloinnin avulla päästään testaamaan aikaisemmin, alueella jossa kustannukset muutoksessa ovat vähäiset, ts. laitteita ei ole vielä ostettu.



**Kuva 42.** Simuloinnin hyödyt muutuskustannuksissa. [Delfoi 2011]

Simulointi on yleensä iteroituva prosessi. Se tarkoittaa, että simulointi ja sitä edeltävä mallintaminen joudutaan usein toistamaan monta kertaa ennen kuin saadaan ratkaisu reaali maailman ongelmaan. Lisäksi voidaan joutua liikkumaan toistuvasti kahden simuloinnin ja mallintamisen avainprosessin välillä ennen kuin päästään seuraavaan avainprosessiin. Kuvassa 43 on kuvattu simuloinnin (tietokonemalli/kokeileminen) ja mallintamisen (käsitteellinen malli ja sen rakentaminen) avainprosesseja. [Robinson 2004]



**Kuva 43.** Simuloinnin ja mallintamisen avainprosessit [Robinson 2004]



### 2.9.1. Robotin etäohjelmointi

Offline-ohjelmointi (Offline Programming, OLP) tai etäohjelmointi on tuotantoa keskeyttämätön robotin ohjelmointimenetelmä, jossa robotin ohjelma tehdään tuotannon ulkopuolella PC-tietokoneella. Menetelmä vastaa CNC-koneiden ohjelmointikäytäntöä, CAD/CAM.

Uusia etäohjelmointimenetelmiä ovat esimerkiksi lisätyn todellisuuden (Augmented reality) menetelmä, jossa virtuaalirobotti lisätään robottisolun 3D-kuvaan. Robotin kinemaattinen malli on siis toisin sanoen lisätty oikeaan soluun ja sillä voidaan tehdä törmäys- ja ulottuvuustarkasteluja ilman että koko solu täytyy mallintaa. [Gîrbacia 2011].

Etäohjelmointi voidaan jakaa tekstipohjaiseen ja mallipohjaiseen ohjelmointiin. Tekstipohjaisessa ohjelmoinnissa ohjelma koodataan tekstieditorilla ja siirretään robotille muistikortin tai tietoverkon välityksellä. Mallipohjainen etäohjelmointi tarkoittaa ohjelmointia simulointi- ja 3D-mallien avulla. Mallin piirteitä voidaan käyttää liikeratojen luomisessa. Yleinen työjärjestys: mallinnus, kalibrointi, ohjelmointi, simulointi, siirto robotille ja lopuksi testaus.

#### **Perinteinen työnkulku robottisolujen ohjelmoinnissa:**

1. Tuote ja kiinnitin suunnitellaan, sekä prototyypit valmistetaan.
2. Vasta tämän jälkeen robottia pääsee ohjelmoimaan.
3. Ongelmiin reagointi vaikeaa ja kallista.

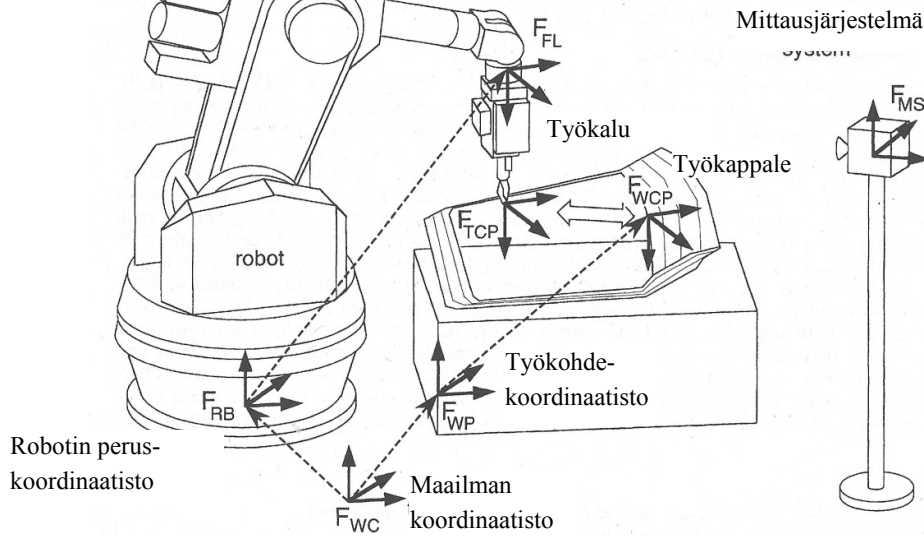
#### **Etäohjelmointi apuna kehitystyössä:**

1. Robotin ohjelmointi alkaa jo kiinnitinsuunnitteluvaiheessa.
2. Ohjelmointi jatkuu robottiohjelmoijan toimesta valmiiksi asti jo ennen kuin ensimmäistäkään tuote- tai kiinnitinprototyyppiä on edes valmistettu.
3. Kun koneet tulevat tuotantotilaan alkaa valmistus eikä ohjelmointi.
4. Asetusajat tuotemuutoksissa putoavat murto-osaan (ohjelmat valmiina).

### 2.9.2. Kalibrointi

Kalibroinnin tarkoituksena on asettaa simulointimalli vastaamaan reaaliympäristöä (*robottisolun kalibrointi*) ja toisaalta asettaa robotin kinematiikan parametrit vastaamaan

tarkasti fyysistä rakennetta (*robottikalibrointi*) sekä *työkalukalibrointi* työkalun tarkkaan määrittämiseen. Tarkoissa töissä kalibrointi on erittäin tärkeä toimenpide. Etäohjelmoidut paikoitukset eivät toteudu jos malli ei vastaa riittävällä tarkkuudella reaali-maailmaa. Kuvassa 44 on esitettyä robottisolun liittyvät komponentit ja niihin liittyvät koordinaatistot. [Schröer 1999].



**Kuva 44.** Robottisolun koordinaatistot. [Schröer 1999]

Robotin ohjaus perustuu koordinaatistojen ja siirtomatriisien käyttöön, joiden avulla paikoitukset (paikka ja asento) voidaan ilmoittaa eri koordinaatistojärjestelmissä. Kuvan 44 tapauksessa koordinaatistomuunnokset voidaan esittää seuraavan kaavan avulla. T tarkoittaa koordinaatistojen välistä siirtomatriisia.

$${}^{WC}_{RB}T \cdot {}^{RB}_{FL}T(q) \cdot {}^{FL}_{TCP}T = {}^{WC}_{WP}T \cdot {}^{WP}_{WCP}T \quad [13]$$

Ohjaus laskee kaavan 1 vasemmalta puolelta työkalupisteen (TCP) paikan solun peruskoordinaatiston (WC) ja oikealta puolelta työkappaleen (WCP) koordinaatiston solun peruskoordinaatistoon (WC). Kun oloarvo (TKP) ja ohjearvo (paikoitus työkappalekoordinaatistossa) on samassa koordinaatistossa voi ohjausjärjestelmä suorittaa robotin liikekäskeyjä. Koordinaatistojen määrittelyissä olevat virheet ilmenevät virheellisinä paikoituksina. Seuraavaksi käsitellään eri virheiden lähteitä koordinaatistoketjussa.

Työkalupiste on määritelty työkalulaipan suhteen  ${}^{FL}_{CP}T$ . Vaikka työkalumäärittäminen olisi tehty tarkasti, työkalun kuluminen, törmäykset ja työkalun vaihto edellyttävät uudelleenmäärittämistä.

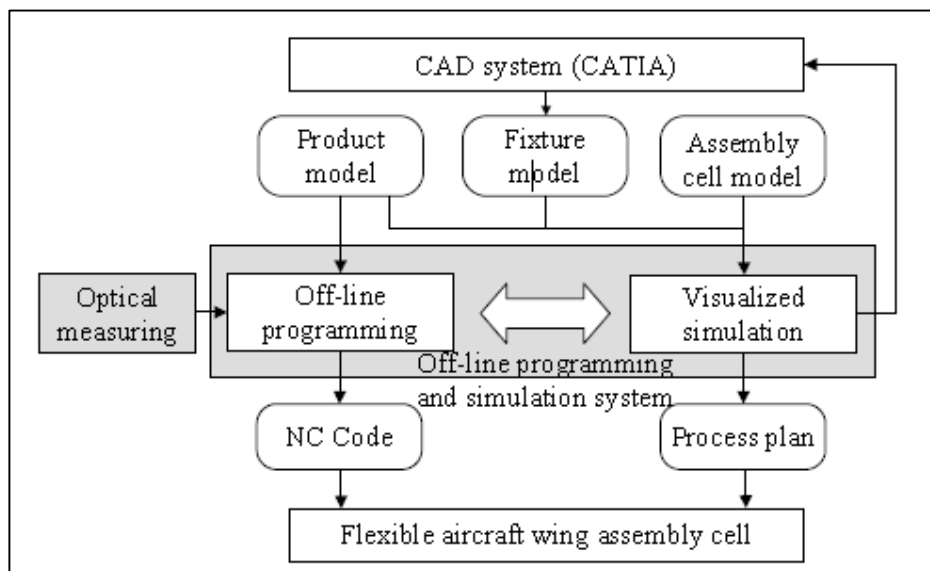
Työkalulaipan koordinaatiston paikka ja asento on määritelty robotin sisäisen kinematiikan avulla peruskoordinaatistoon  ${}^{RB}_{FL}T(q)$ . Kaavassa oleva  $q$  ilmaisee siirtomatriisin riippuvuuden robotin nivelkulmista. Nivelkulman muuttuessa siis siirtomatriisikin muuttuu. Virheitä syntyy todellisen rakenteen ja matemaattisen mallin välisistä eroista, nivelien joustoista ja välyksistä, vaihteistojen epätarkkuuksista sekä nivelvarsien tai robottijalustan muodonmuutoksista.

Robotin peruskoordinaatisto on kuvattuna solun koordinaatistoon  ${}^{WC}_{RB}T$ . Robotin paikka ja asento solussa tulee olla tarkasti määritetty. Jos robotti on lineaariradalla tai muulla liikkuvalla alustalla, myös siirtomatriisi muuttuu robotin liikkuesssa.

Työkappalekoordinaatisto (WCP) on kuvattu työkohdekoordinaatiston (WP) kautta solun peruskoordinaatistoon  ${}^{WC}_{WP}T \cdot {}^{WP}_{WCP}T$ . Työkohde voi olla työpöytä, jigi, kääntöpöytä tms. johon koordinaatisto on kiinnitetty. Työkohde- ja työkappalekoordinaatistoja käytetään usein etäohjelmoinnissa, kun 3D- mallin piirteitä käytetään ratojen määrittämisessä. Toisaalta kalibroidussa solussa voidaan käyttää vaikka robotin peruskoordinaatistoa referenssinä, koska työkappaleen tarkka paikka on määritelty. Työkappalekoordinaatisto määritetään 3D- malliin siten, että se on myös helppo määrittää reaalikappaleeseen. Kalibroidussa solussa voi radat luoda vapaasti määritetyssä koordinaatistossa ja siirtää käsiohjaimen kautta koordinaatistodata oikealle robotille. Käyttäjäkoordinaatisto voidaan määrittää vaihtoehtoisesti työkappaleen kiinnittimeen tai pöytään ja asemoida työkappale tarkasti samaan paikkaan kuin mallissa, eikä työkappalekoordinaatistoa aina tarvitse käyttää. Vaikka em. koordinaatistot olisivatkin tarkasti määritetty, paikoitusvirheitä saattaa syntyä esimerkiksi työkappaleen dimensioiden vaihdellessa kappaleesta toiseen. Yksi vaihtoehto on käyttää ulkoista mittalaitetta, jonka mittausrvojen avulla koordinaatiston paikoitusta muutetaan aina työkappaleen vaihtuessa. Näin kaikki ko. koordinaatistoon määritetyt paikoitukset siirtyvät mukana.

Kuvassa 44 on esitetty vielä mittalaitteen koordinaatisto ( $F_{MS}$ ) mikä myös täytyy määrittää solun peruskoordinaatiston suhteen. [Schröer 1999]

Erilaiset tehtävät vaativat kuitenkin erilaisia kalibrointityökaluja ja menetelmiä. Esimerkiksi joustavilla työkaluilla voidaan kompensoida etäohjelmoidun radan poikkeamia ja näin ohjelman tarkkuusvaatimus pienenee jouston suunnassa. Jos prosessiin rakennetut joustoelementit eivät riitä kompensointiin, ohjelmoitua rataa on pystyttävä muuttamaan työvaiheen aikana. Tällöin on käytettävä muita menetelmiä, ongelma ei ole silloin kalibrointiongelma. Konenäkö- ja mittausjärjestelmien käyttö auttaa paikoitus- ja mittavirheiden kompensoinnissa. Esimerkiksi lentokoneteollisuuden tarpeisiin on kehitetty siipien kokoonpanoon niittien porauksen joustava etäohjelmointiympäristö. Siinä käytetään optista mittalaitetta, jonka avulla tuotantomallia korjataan vastaamaan oikeaa työkalupalletta. [Cheng 2011]



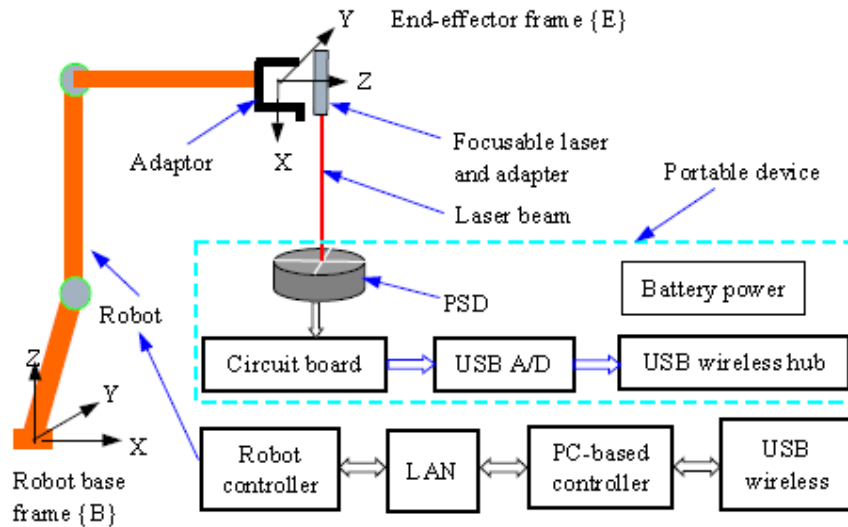
**Kuva 45.** Simuloinnin ja mallintamisen prosessi joustavassa siipikokoonpanossa. [Cheng 2011]

Kuvassa 45 on järjestelmäkuvaus joustavasta siipikokoonpanosta. Prosessissa on käytetty etäohjelmointia ja simulointia rinnakkain. Optisella mittauksella on korjattu mallin virheet ennen ohjelmointia. Simulointimallilla testataan ohjelma paikoitusten ja törmäyksien varalta. [Cheng 2011]

Robotteja käytetään usein mittalaitteena määritettäessä työkalu- ja työkohdekoordinaatistoja. Se onkin helppo tapa saada koordinaatistojen paikoitustiedot etäohjelmointiympäristöön. Vaihtoehtoisesti voidaan käyttää kalibrointiohjelmia, joiden avulla malli ja oikea ympäristö saadaan vastaamaan toisiaan. Robotilla mittaaminen edellyttää että oikean robotin oma kinemaattinen malli on kunnossa, ts. robotti itsessään on kalibroitu. Lisäksi täytyy varmistaa, että etäohjelmointiympäristössä oleva robottikinematiikka vastaa todellista robotin kinematiikkaa, että on siis valittu oikea malli. Mallin robotin ei tarvitse vastata täydellisesti käytettävää robottia. Peruskinematiikka riittää osoittamaan konfiguraatiot ja dynaamisen mallin liikeradoille. Tarkat paikoitukset lasketaan ympäristön koordinaatistojen avulla. Jos paikoitustieto muodostettaisiin nivelarvoina, tulisi robottien kinematiikkojen vastata täydellisesti toisiaan. Se vaatisi valtavan määrän kalibrointeja ja dynaamista mallintamista, eikä toisi kuitenkaan mitään lisäarvoa liikeratojen ohjelmointiin.

Roboteissa itsessään on useita virhelähteitä, kuten vaihteistojen epätarkkuus, lämmöstä ja voimista johtuvat muutokset. Yongin ja Ningin mukaan merkittävin virhelähde löytyy robotin kinematiikan parametrien virheistä. Kinematiikan kalibrointimenetelmiä robotin tarkkuuden parantamiseksi on kehitetty ja tutkittu laajasti. Usein on kuitenkin päädytty monimutkaisiin menetelmiin, jotka ovat kalliita ja hankalia käyttää tehdasolosuhteissa. Robotin valmistajat kalibroivat robotit ennen toimitusta ja yleensä kinemaattinen rakenne ei ole muuttunut käyttöönoton jälkeen. Muissa tutkimuksissa on todettu kuitenkin, että 90 prosenttia robottien paikoitusvirheistä johtuvat nivelen offset- parametrin virheestä. Nivelen offsetilla tarkoitetaan robotin ohjaimeen syötettyä korjausarvoa nivelkulman mittaukselle. Parametrin avulla korjataan nivelkulmavirhettä, eli nivelkulman mittauksen ja todellisen kulman eroa. Pienikin virhe kulman mittauksessa aiheuttaa selkeän poikkeaman paikoituksessa. Yong ja Ning kehittivät laseriin ja PSD- elementtiin perustuvan kalibrointimenetelmän niveloffsetin kalibroimiseksi. Menetelmä perustuu yhden paikoituspisteen avulla tehtävään kalibrointiin ja on halpa (PSD-anturin hinta on n.100 euroa), siirrettävä, sekä automatisoitavissa, kuva 46 [Yong 2011]

PSD (Position Sensitive Detector) on valoherkkä anturi, joka ilmaisee siihen kohdistetun lasersäteen paikan jopa  $0.1\mu\text{m}$ :n tarkkuudella. Järjestelmä koostuu robotin työkalulaippaan kiinnitetystä laseryksiköstä ja pöydälle asennetusta PSD- elementistä.



**Kuva 46.** Nivel-offsetin kalibrointijärjestelmä. [Yong 2011]

Menetelmässä robotti paikoitetaan eri nivelvarsien asennoissa tarkasti samaan paikoitukseen lasersäteen ja PSD-anturin avulla. Nivelkulma-arvot tallennetaan jokaisessa paikoituspisteessä ja kulmavirheet ratkaistaan kokonaissiirtomatriisin avulla. Kun tunnetaan robotin ohjaimen käyttämä kokonaissiirtomatriisi, voidaan laskea sisäisen kinematiikan ehdottamat nivelkulmat ko. paikoituspisteisiin. Näitä arvoja verrataan todellisiin nivelkulma-arvoihin ja nivelen offset- parametrin arvo saadaan ratkaistua. Lopuksi ratkaistu nivelen offset-parametri korjataan robotin ohjausjärjestelmässä. [Yong 2011]

Jäysteenpoistosovellukset sisältävät normaalisti erilaisilla joustoilla kompensoitavia mitta- ja paikoitusvirheitä. Nämä poikkeamat ovat moninkertaisia nivelen offset-parametrin virheeseen verrattuna, joten yleensä ei ole tarvetta kalibroida robotin kinematiikkaa.

### 2.9.3. Ohjelmistot

Etäohjelmoinnista (OLP, Offline Programming) on tulossa perusvaatimus moniin robotisovelluksiin, varsinkin jäysteenpoistoon ja viimeistelyyn. Hitsauksessa etäohjelmointia on käytetty jo pitkään. Markkinoilla on useita robottien etäohjelmointiohjelmistoja, joilla kaikilla on omat erityispiirteensä. Useat ovat robottivalmistajan omia, joten niillä voidaan ohjelmoida vain yhtä robottimerkkiä (esim. Fanuc/Roboguide, ABB/Robotstudio). Yleiskäyttöinen, kaikille roboteille sopiva ohjelmisto on esimerkiksi

Delmia V5, joka on erittäin iso ohjelmisto. Ohjelman monipuolisuus tosin hankaloittaa sen käyttöä pienissä sovelluksissa ja edellyttää laajahkoa koulutusta.

Offline ohjelmistojen välillä on eroja hinnan, ominaisuuksien ja laajuuden suhteen. Käyttöliittymätoteutukset vaihtelevat, samoin prosessikohtaiset ohjelmointi- ja simulointityökalut sekä kalibrointityökalut.

#### **2.9.4. Fastsimu- etäohjelmointiohjelmisto**

Fastems Fastsimu® on tehokas offline- ohjelmointi- ja simulointiohjelma, joka on erityisesti suunniteltu robotisoituun jäysteenpoistoon ja viimeistelyyn. Fastsimu on suunniteltu helppokäyttöiseksi ja nopeasti ohjelmoitavaksi. Käyttöliittymä on pyritty rakentamaan siten, että se sisältää vain työhön vaadittavat työkalut. Fastsimu pohjautuu Visual Componentsin luomaan ohjelmistoalustaan. Tavoitteena on ollut oleellisten teknisten toimintojen pakkaaminen mahdollisimman helppokäyttöiseen muotoon ja samalla käyttöliittymän tekeminen selkeäksi ja yksinkertaiseksi. Lisäksi Fastsimu on robottimerkki-riippumaton ohjelmisto. Fastsimussa on erikoistyökalut viimeistelyprosesseissa tyypillisten pitkien ja monimutkaisten työstöratojen käsittelyyn. Kuvassa 47 on näkymä Fastsimun käyttöliittymästä. [Fastems 2012]

Ohjelmisto voidaan toimittaa uuden jäysteenpoistosolun mukana, jolloin ympäristö on valmiiksi luotuna. Käyttäjän ei tarvitse lisätä etäohjelmointiympäristöön mitään muuta kuin työkappaleen ja kiinnittimet. Kaikki soluun liittyvät oheislaitteet ja työkalut ovat valmiina kirjastossa. Toisaalta on paljon olemassa olevia soluja, joihin etäohjelmointiympäristö toisi uudenlaista tehokkuutta. Tällöin täytyy solu luoda alusta alkaen, eikä kaikkia työkaluja ja oheislaitteita löydykään suoraan kirjastoista. Tällöin käyttäjä ostaa työn palveluna tai hän voi luoda ympäristön itse.

Ohjelmiston ominaisuudet: [Fastems 2011]

- laaja tuki CAD mallien sisäänlukuun. Perusformaatit: VRML, STL, DXF, 3ds  
Optiona: Inventor, JT, Open, Parasolid, SolidWorks, CATIA, STEP, ProE, IDEAS NX, IGES.
- avoin robottimerkkiriippumaton ohjelmisto
- mahdollisuus hyödyntää julkisia komponenttikirjastoja solun mallintamisessa

- pyörityspöytien ja lineaariratojen käyttö robotin ulkoisina akseleina
- kattavat työkalut simulointimallin kalibrointiin
- robottiohjelmien luonti automaattisesti CAD/CAM –ohjelmistolla tehdyistä työstöradoista, sisäänluku APT- tai NCI-formaateissa
- robotin paikoituspisteiden luonti automaattisesti 3D CAD-mallin piirteitä hyväksikäyttäen
- monipuoliset paikoituspisteiden ja ratojen muokkaustoiminnot – paikan ja asennon säätö, pistejoukkojen kopiointi, siirto ja peilaus
- prosessiparametri- ja I/O –määrittelyt, tuki sovelluskohtaisille käskyille.

### Simulointi:

- törmäys- ja ”läheltä piti”-tarkastelut
- robotin nivelrajojen, -nopeuksien ja -kiihtyvyyksien valvonta sekä singulariteetin tunnistus
- työkaluvaihtojen simulointi
- tahtiaikatarkastelu
- kaksisuuntaiset kääntäjät robottiohjelmien siirtämiseen Fastsimun ja robottiohjaimen välillä
- Python-ohjelmointikieli edistyneille käyttäjille omien lisäominaisuuksien toteuttamiseksi.



*Kuva 47. Fastsimun käyttöliittymä ja simuloitu solu.*



Tavoitteena on helpottaa konepajojen siirtymistä etäohjelmointiin robottisovelluksissaan. Kohderyhmänä erityisesti lastuava metalliteollisuus, jäysteenpoisto ja hiontasovellukset. Tuote sopii myös oppilaitosten käyttöön robotiikan opetuksessa. Ohjelman kehityksessä tavoitteena on ollut tehdä helposti opittava ja helppokäyttöinen etäohjelmointiohjelmisto. [Fastems 2011]

## 2.10 Robottiturvallisuus

Robotti on turvallisuuden kannalta ongelmallinen kone, koska monimutkaisena automaatiolaitteena se saattaa liikkua ennalta aavistamattomasti laajassa kolmiulotteisessa tilassa. Nopeudet ja voimat ovat usein niin suuria, että robotin iskemäksi tai puristamaksi joutuminen aiheuttaa kuoleman tai vakavan vamman. Lisäksi robotteihin liittyy mm. seuraavia vaaratekijöitä [Siirilä 2009]:

- Robottia opettamalla (käsiohjelmointilaitteella liikuttamalla) ohjelmoitaessa voi samassa tilassa olla koneita, jotka käynnistyvät aiheuttaen tapaturmavaaran.
- Tarraimesta robotin liikkeen aikana irtoava kappale voi aiheuttaa vamman. Lisäksi suurnopeuskoneistuksen työkalut voivat rikkoutuessaan aiheuttaa vakavan vahingon.
- Ohjelmoinnissa voidaan tehdä näppäilyvirheitä, joiden seurauksena robotin liikerata muuttuu huomattavasti.
- Robotin sisäiset ohjelmistoviat, sähköiset häiriöt, alajärjestelmien venttiiliviat tai -vuodot voivat aiheuttaa virheellisiä toimintoja tai muita ongelmia.

Robotti on osa konejärjestelmää. Kun useampi kone tai koneiden tietyt alueet on suunniteltu toimimaan yhdessä, ne on suunniteltava ja rakennettava siten, että pysäytysohjaimet, hätäpysäytyslaitteet mukaan lukien, pysäyttävät kyseisen koneen lisäksi kaikki koneeseen yhteydessä olevat laitteet, jos laitteiden toiminnan jatkaminen voi aiheuttaa vaaraa [Siirilä 2009]

Robottiturvallisuutta koskevat seuraavat standardit:

- SFS-EN ISO 11161 + A1 2010 Koneturvallisuus. Valmistusjärjestelmien koneyhdistelmät. Perusvaatimukset
- SFS-EN ISO 14121-1 2007 Koneturvallisuus. Riskin arviointi. Osa 1: periaatteet
- SFS-ISO/TR 14121-2 2010 Koneturvallisuus. Riskin arviointi. Osa 2: käytännön opastusta ja esimerkkejä menetelmistä
- SFS-EN 953 + A1 2009 Koneturvallisuus. Suojukset. Kiinteiden ja avattavien suojusten Suunnittelun ja rakenteen yleiset periaatteet
- SFS-EN 349 + A1 2008 Koneturvallisuus. Vähimmäisetäisyydet kehonosien puristumisvaaran välttämiseksi

- SFS-EN ISO 13857 2008 Koneturvallisuus. Turvaetäisyydet yläraajojen ja alaraajojen ulottumisen estämiseksi vaaravyöhykkeille
- SFS-EN ISO 10218-1 Teollisuusrobotit. Turvallisuusvaatimukset. Osa 1: Robotti
- ISO-EN 10218-2 Robots for industrial environments, Safety requirements, Part 2 Robot systems and integration.

Robottisolun valmistajan täytyy noudattaa seuraavaa menettelyä saattaakseen sen turvallisuuden kannalta lainvoimaiseksi.

Koneen valmistajan tehtävä on [Työsuojeluhallinto 2009]

- arvioida riskit
- selvittää konetta koskevat turvallisuusvaatimukset
- suunnitella ja rakentaa kone olennaisten turvallisuusvaatimusten mukaisesti
- laatia käyttöohjeet ja tehdä koneeseen tarvittavat merkinnät
- laatia tekninen tiedosto
- tehdä vaatimustenmukaisuusvakuutus
- kiinnittää koneeseen CE-merkintä.

Valittujen suojalaiteratkaisujen osalta tulee noudattaa niille tehtyjä standardeja. Jos jokin tapaturma kuitenkin tapahtuu, voidaan dokumentein osoittaa että turvamääräyksiä on noudatettu ja mahdollisia riskejä on yritetty pienentää.

### 3 ROBOTIIKAN OPETUS

Robottiikan opetusta annetaan ammatillisissa oppilaitoksissa, kuten ammattiopistoissa ja aikuiskoulutuskeskuksissa, sekä ammattikorkeakouluissa ja yliopistoissa. Tämän lisäksi robottien maahantuojat järjestävät koulutusta maahantuomiensa robottien käytön tueksi. Oppilaitokset käyttävät perusopetukseen yleensä yhtä robottimerkkiä. Vaikka robottien perustekniikka ja rakenne on samankaltainen, on ohjauksissa kuitenkin merkittävästi eroavaisuuksia. Usean eri merkkisen robotin opettaminen yhtäaikaaisesti ei ole tehokasta. Robotiikan opetus jää tällä hetkellä perusasioiden läpikäymiseksi, eikä varsinaisia robottiammattilaisia kouluteta oppilaitoksissa, vrt. koneistajia. Lisääntyvä automaatio kuitenkin edellyttää robotiikan ja lisäksi etäohjelmoinnin työkalujen tuntemista.

Tämän työn kannalta tavoitteena on löytää menetelmiä robotiikan tehokkaaseen opetukseen ja toisaalta löytää käytännönläheinen lähestymistapa robottisolun etäohjelmoinnin käyttöönottoon. Opetusmenetelmän kehityksessä etsitään ensin ne tekijät mitkä on ymmärrettävä, jotta robotin ohjelmointi olisi mahdollista. Tämän jälkeen luodaan harjoitteet robotin peruskäyttöön. Toisessa vaiheessa otetaan etäohjelmointiympäristö mukaan. Tavoitteena on löytää ja opettaa välttämättömimmät tekijät mitkä ovat tarpeellisia yksinkertaisen mallisolun hallitsemiseksi.

#### 3.1 Robotiikan opetusmenetelmät ja sisältö

Robottiikan opetusmenetelmät jakaantuvat teoriaopetukseen ja käytännön harjoitteluun. Teoriaopetus rajautuu yleisrobotiikkaan ja robottisovelluksiin. Yleisesti käydään läpi teollisuusrobottien rakenteet, tehtävät ja työkalut. Solusuunnittelusta ja robotin hankinnasta käydään perusteet läpi. Korkeakouluissa opetetaan syvällisemmin robotin kinemaatiikkaa kuin ammattikorkeakouluissa. Ammattiopistotasolla robottikoulutus sisältää usein vain käyttäjäkoulutuksen, puuttumatta tarkemmin robotin toimintaan.

Harjoittelu oikeilla laitteilla, niin sanottu laboraatioharjoittelu on nykyisin kiinteä osa monien oppilaitosten opiskelua. Tiedetty tosiasia on, että harjoittelu todellisessa

ympäristössä, työssä oppiminen, ja harjoittelu oikeilla laitteilla todelliseen elämään kuuluvilla laitteilla tehokkainta mahdollista valvottua opettamista työelämää varten. [Vuorinen 1993]

Käytännön harjoittelut ovat yleensä merkkiriippuvaisia, jokaisella robottimerkillä on erilainen ohjelmointikieli. Koska oppilaitoksilla ja myös maahantuojiin kouluttajilla on yleensä kerrallaan käytössä vain muutama robotti, ryhmät ovat pieniä. Ryhmäkoon kasvassa menetetään opetuksen tehokkuutta. Tällöin joudutaan harjoittelemaan osaa asioista jollakin muulla keinoin kuin oikeilla laitteilla, esimerkiksi asennus, suunnittelu yms. harjoitteluja tehden.

Simulointi tarkoittaa sitä, että tapahtuma tai prosessi jäljitellään joko toistamalla tai ennakoimalla todenmukaisena jäljiteltäviä prosessia. Tällöin opiskelua varten rakennetaan mahdollisimman pitkälle todellista tilannetta vastaava ympäristö, sen osa tai muu opiskeltava ilmiö. Simuloinnin tarkoituksena on jonkin monimutkaisen tilanteen tutkiminen ja siihen liittyvien taitojen harjoittelu ilman todellisessa tilanteessa olevia epäonnistumisen paineita tai todellisia vaaroja. [Vuorinen 1993]

Etäohjelmointiympäristöt ja simulointiohjelmistot mahdollistavat käytännön harjoittelun simuloimalla. Simulointi tuo täysin uusia mahdollisuuksia koulutukseen. Ongelmana on usein simulointiohjelmistojen monimutkaisuus, joiden opiskeluun ei koulutusohjelmaan varattu aika riitä. Toinen merkittävä ongelma liittyy ohjelmistojen lisenssisopimukseen. Opiskelijan tulisi pystyä käyttämään ohjelmistoa myös oppilaitoksen ulkopuolella, koska ohjelmistojen opettelu vaatii aikaa ja motivoituneen tunnetilan. Jos ohjelmiston lisenssi ei salli etäkäyttöä, on suuri riski että ohjelmiston käyttö jää vähäiseksi, eikä syvällistä oppimista tapahdu. Toisaalta oikean robotin käyttöharjoituksia ei korvaa mikään simulaattori. Ympäristölaitteet ja varsinainen prosessi tuo omat muuttajat sovelluksen toimintaan. Kehittyneet simulaattorit ovat kuitenkin hyödyksi myös koulutuksessa ja ne antavat mahdollisuuden kehittää aivan uusia etäoppimisympäristöjä.

Maahantuojiin järjestämät kurssit ovat usein jaoteltu käyttäjäryhmän ja vaatimustason mukaisesti. Yleensä niin että edelliset kurssit tulee olla suoritettuna kun seuraavalle kurssille osallistuu. Peruskäyttäjille, huoltohenkilökunnalle ja sovellussuunnittelijoille

on omat tarkoituksenmukaiset kurssinsa. Kehiteltävä ympäristö sopii kaikenlaisen robotiikan opetukseen.

### 3.2 Robotiikan peruselementit

Robotiikan peruselementit tarkoittavat tässä yhteydessä niitä elementtejä, joiden avulla robotin käynnistys, sammutus, opettamalla ohjelmointi tai etäohjelmointi voidaan turvallisesti suorittaa. Peruskäyttäjän tulee saada käsiohjaimen avulla robotti liikkumaan hallitusti, paikoitukset tallennettua, ohjelma testattua ja käynnistettyä. Etäohjelmoijan on näiden lisäksi hallittava konfiguraatiot ja koordinaatistot tarkemmin. 3D-mallinnuksen osaaminen on iso etu etäohjelmointiympäristöä rakennettaessa ja muokkaamisessa.

Peruskäyttö:

- robottiturvallisuus (turva-aidat, portit, turvareleet, käyttötilanteiden riskit)
- käsiohjain (sallintalaite, painikkeiden toiminta)
- koordinaatistojen käyttö ja luonti (perus-, käyttäjä- ja työkalukoordinaatistot)
- robotin TKP:n liikuttelu hallitusti eri koordinaatistoissa
- ohjelmointi (liikekäsky, aliohjelmointi, ehtolauseet, I/O-komennot, paikoituksen määrittely)
- testaus ja ohjelman käynnistys (automaattiajo, käsiajo, hidastettu nopeus)
- tiedonsiirto (ohjelman tallennus ja lataus robotille)

Etäohjelmointi:

- robottikinematiikka (konfiguraatio ja kinemaattiset tehtävät)
- 3D- mallinnus (kappaleiden siirto, liittäminen, kierto)
- ympäristön kalibrointi (malli ja oikea solu vastaamaan toisiaan)
- paikoituspisteiden luonti (pisteiden siirto, kierto, määrittely)
- simulointi ja testaus (nopeudet, tarkkuus, törmäykset, prosessiosaaminen)
- postprosessointi (ohjelman muutos robotin ymmärtämään muotoon)

### 3.3 Jäysteenpoiston peruselementit

Jäysteenpoiston robotisointiin liittyy seuraavat tekijät:

- jäysteen määrittely (muoto, koko, materiaali ja sijainti)
- menetelmän valinta (kiinteä vai robotissa oleva työkalu/työkalut)
- työkalun valinta (koneviila, flap grinder, harja, kärkiupotin)
- työkalujouston valinta (varsinaisen työkalun jousto, työkalupitimen jousto, voimaohjaus) tai ohjelmanaikainen radankorjaus
- ohjelmointiympäristön valinta (opettamalla, etäohjelmointi)
- joustoyksikön valinta (passiivinen, aktiivinen järjestelmä)
- turvallisuustarkastelu (ottaen huomioon valittujen menetelmien erityispiirteet).

Kaikkia menetelmiä ja niiden yhdistelmiä tulee vertailla systemaattisesti, jotta paras lopputulos teknisesti ja taloudellisesti saavutettaisiin.

## 4 OPPIMISYMPÄRISTÖ

Tässä kappaleessa käydään läpi oppimisympäristön kehitystyö. Se sisältää opetuslaitteen esittelyn, testiympäristön teon vaiheet ja testauksen. Lisäksi esitetään valitut harjoitteet ja käyttöohjeistukset konseptin ymmärtämiseksi.

Kuten kappaleessa 2.9.4 todettiin, Fastsimua tullaan käyttämään uusissa ja jo olemassa olevissa soluissa. Kun uusi jäysteenpoistosolu toimitetaan etäohjelmointiympäristöineen, ei vaadita samanlaista perehtymistä kuin täysin uuden ympäristön luomisessa. Ainoastaan asiakkaan työkappale ja kiinnittimet mallinnetaan ja siirretään etäohjelmointiympäristöön. Tämän kehitettävän oppimisympäristön lähtökohtana on olemassa oleva robottisolun, jolle luodaan etäohjelmointiympäristö. Tämä tarkoittaa myös työkalujen, robotin jalustan ja pöytien mallintamista ympäristöön.

### 4.1 Oppimisympäristön kuvaus

Ympäristö koostuu teoriaosuudesta ja käytännön harjoitteista. Käytännön harjoitteet on jaettu peruskäyttöharjoituksiin ja etäohjelmointiharjoituksiin. Teoriaosiossa käydään läpi teollisuusrobotin toiminta aina kinematiikan tarkasteluun asti, koska koordinaatistojen ja konfiguraation ymmärtäminen on edellytys etäohjelmointiympäristön luomiseen ja sen tehokkaaseen käyttöön. Kehitetty opetuslaite auttaa käytännönläheisesti kinemaattisten ratkaisujen opetuksessa.

Peruskäyttöharjoituksien tavoitteena on syventää koordinaatistoajattelua ja käsiohjaimen käyttöä. Harjoituksien jälkeen käyttäjä osaa liikutella robottia hallitusti eri koordinaatistoissa ja työkaluilla. Perusohjelmien teko ja editointi kuuluu myös perusharjoituksien sisältöön.



Etäohjelmointiharjoitukset sisältävät Fastsimun käyttöharjoituksia ja ohjeistuksen etäohjelmointiympäristön luomiseksi. Tavoitteena on saada luotua ympäristö esimerkki-en avulla ja sitten oikealla robotilla toimiva robottiohjelma.

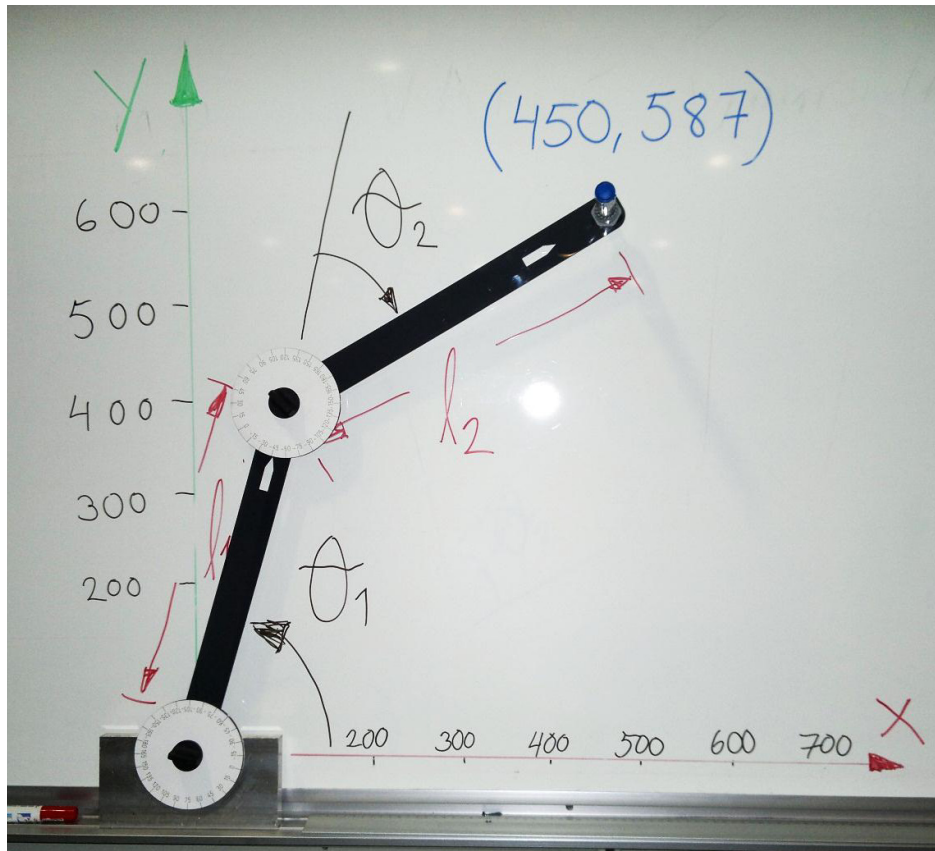
## **4.2 Kinematiikan opetuslaitteisto**

Laitteiston avulla voidaan opettaa havainnollisesti robotin kinematiikan ja koordinaatisten merkitys robotin ohjauksessa. Lisäksi konfiguraation käsite: robotti voi toteuttaa saman paikoituksen usealla eri nivelien yhdistelmillä ja asennoilla. Laitteen avulla voidaan osoittaa matemaattinen yhteys ja sen käyttö robotin ohjauksessa.

- kinematiikka
- interpolointi
- konfiguraatio
- koordinaatistot

### **4.2.1. Laitteiston rakenne**

Laitteisto koostuu kahdesta nivelvarresta, joiden pituudet ovat 401mm, sekä mittasteikosta, kuva 49, josta voidaan lukea nivelen asento. Materiaalina on 3,2 mm laserilla muotoon leikattu PMMA- muovilevy. Työkalupisteeseen voidaan kiinnittää tussi tai liitu, millä voidaan piirtää taululle paikoituksen piste ja myös kuvioita. Laitteella voi havainnollistaa myös liikkeenohjausta ja interpolaatiota. (nivelliike, lineaari- ja ympyränkaari-interpolaatio).



**Kuva 48.** Opetuslaite ja tarvittavat merkinnät.

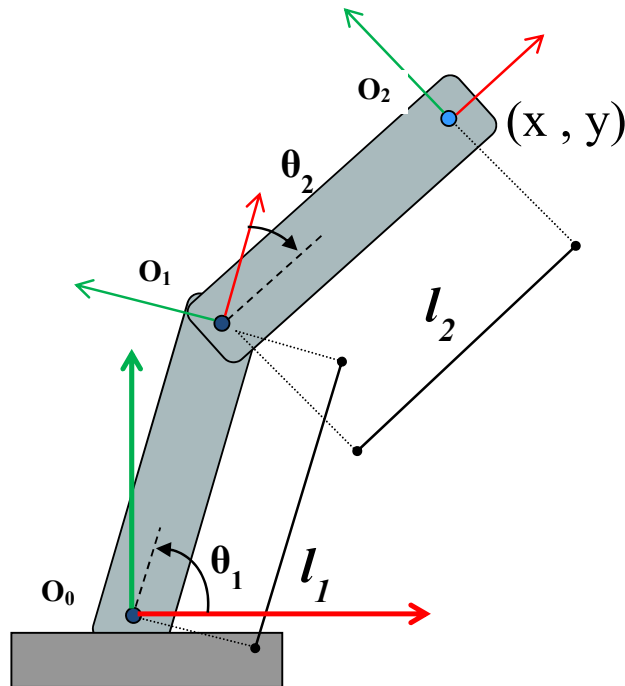
Kuvassa 48 on opetuslaitteisto 2-nivelvartisena versiona. Laitteistoon voidaan tarvittaessa lisätä nivelvarsia. Perusasian läpikäymiseksi kaksinivelinen esimerkki on riittävä. Varteen on kiinnitetty lasermerkattu kulma-asteikko, minkä avulla varren asento voidaan ilmaista.



**Kuva 49.** Kulman osoitinkieppo.

#### 4.2.2. Kinemaattiset ratkaisut

Laitteen avulla voidaan osoittaa kappaleessa 2.2.2 esitetyn Denavit-Hartenberg menetelmän toimivuus käytännössä. Taulukko 1 sisältää kaksinivelisen tasomanipulaattorin D-H parametrit. Kuvassa 50 on kaikki tasomanipulaattorin muuttujat. Kuvaan on piirretty myös järjestelmään liittyvät koordinaatistot: peruskoordinaatisto ( $O_0$ ), varren 1 origo ( $O_1$ ) ja varren 2 origo ( $O_2$ ).



**Kuva 50.** 2-nivelinen tasomanipulaattori x,y- koordinaatostossa.

**Taulukko 1 .** D-H parametrit 2-niveltapauksessa.

Varsi (i)	D-H Parametrit							
		a		$\alpha$		d		$\theta$
1	$L_1$	285	0		0		$\Theta_1$	30
2	$L_2$	285	0		0		$\Theta_2$	25

Kappaleen 2.2.2 mukaan kokonaissiirtomatriisi näyttää yhtälön 14 mukaiselta. Se on lopputulos kun molemmat siirtomatriisit ovat kerrottu yhteen. Se kuvaa opetuslaitteiston työkalupisteen paikan peruskoordinaatistoon. Merkintä  $\cos \theta_{12}$  tarkoittaa  $\cos (\theta_1 + \theta_2)$ .

$${}^0_2T = \begin{bmatrix} \cos \theta_{12} & -\sin \theta_{12} & 0 & L_1 c_1 + L_2 c_{12} \\ \sin \theta_{12} & \cos \theta_{12} & 0 & L_1 s_1 + L_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [14]$$

Kuvassa 51 on kokonaissiirtomatriisi laskentataulukkomuodossa. Se laskee ja näyttää D-H parametritaulukon mukaisilla arvoilla TKP:n paikan x- ja y-koordinaatein ja asennon yksikkövektoriesityksenä.

		kiertomatriisi			siirrosvektori	
		x	y	z		
${}^0_2T$		0,57	-0,82	0,00	410,29	X
		0,82	0,57	0,00	375,96	Y
		0,00	0,00	1,00	0	Z
		0	0	0	1	

**Kuva 51.** Excel-laskentataulukko 2-nivel tapauksessa.

Excel-taulukko yhdessä laitteiston kanssa kuvaa yksinkertaistettuna robotiikan perustehtävän. Kinemaattinen malli, jonka avulla voidaan matemaattisesti laskea robotin työkalupisteen paikka annettujen nivelkulmien avulla, eli suora kinemaattinen tehtävä.

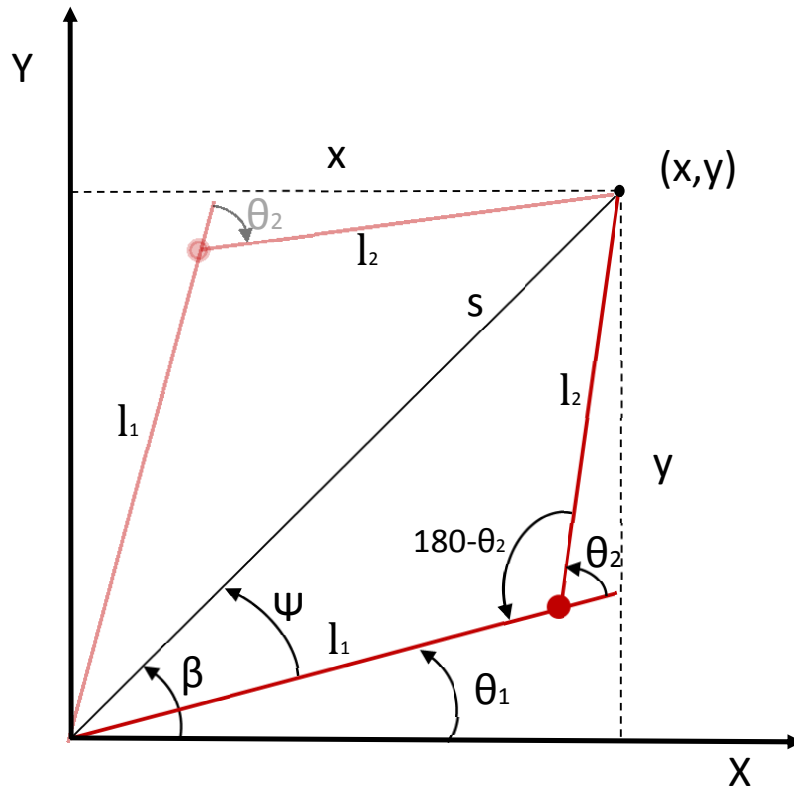
Käänteisen kinemaattisen tehtävän suorittaminen ei olekaan niin yksinkertaista ja suoraviivaista. Käänteisessä tehtävässä ohjauksen tulee laskea tavoitepisteen vaatimat nivelkulma-arvot jokaiselle nivelelle. Opetuslaitteen avulla ratkaistaan seuraavaksi 2-nivelisen tasomanipulaattorin käänteinen kinemaattinen tehtävä algebrallisella ja geometrisella tavalla. Algebraalinen tapa perustuu kokonaissiirtomatriisiin, erityisesti D-H-menetelmään.

Ratkaisu ei ole yksikäsitteinen, vaan manipulaattori voi toteuttaa paikan kahdella eri nivelien asennoilla, eli konfiguraatioilla. Huomioitavaa on että tämän manipulaattorin vapausasteet eivät riitä asennon toteuttamiseen kahdella konfiguraatiolla, ainoastaan paikan. Lisäksi on tarkasteltava onko ratkaisua olemassa, eli onko paikoitus robotin

työalueella ja riittääkö vapausasteet toteuttamaan haluttu orientaatio. Usein löytyy vaihtoehtoisia ratkaisuja, joista ohjausjärjestelmän tulee valita sopivin. [Goldenberg 2007]

### Geometrinen ratkaisu

Kuvassa 52 on 2-nivelisen tasomanipulaattorin kaksi ratkaisua ja apuviiva  $s$ . Tämä on geometrisen ratkaisun pohjana.



**Kuva 52.** Kaksinivelisen mekanismin muuttujat ja konfiguraatiot. [Craig 2005]

Ratkaisussa aloitetaan kulman  $\theta_2$  ratkaisemisesta. Käytetään kosinilausetta,

$$c^2 = a^2 + b^2 - 2ab \cos C \quad [15]$$

Minkä jälkeen arvot siirretään kaavaan, josta voidaan kulma ratkaista.

$$(x^2 + y^2) = l_1^2 + l_2^2 - 2l_1l_2 \cos(\theta_2) \quad [16]$$

On huomioitava että  $\theta_2$  voi saada kaksi arvoa, positiivinen ja negatiivinen, riippuen manipulaattorin konfiguraatiosta. Lopullinen laskukaava  $\theta_2$  :lle :

$$\theta_2 = \pm \arccos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right) \quad [17]$$

Seuraavaksi ratkaistaan kulma  $\theta_1$  . Ensin tulee ratkaista kulma  $\beta$  sekä apukulma  $\psi$ . Kuvasta 52 nähdään mistä nämä kulmat muodostuvat.  $\beta$ - kulma ratkeaa kahden argumentin argustangentin avulla:

$$\beta = \arctan 2\left(\frac{y}{x}\right) \quad [18]$$

Sinilauseen avulla avulla ratkeaa apukulma  $\psi$  :

$$\frac{\sin B}{b} = \frac{\sin C}{c} \quad [19]$$

Sijoitetaan kuvasta arvot, huomioi että sinifunktion arvo on 180 asteen kerrannainen, joten voidaan kirjoittaa seuraavasti :

$$\frac{\sin \psi}{l_2} = \frac{\sin(180 - \theta_2)}{\sqrt{x^2 + y^2}} = \frac{\sin(\theta_2)}{\sqrt{x^2 + y^2}} \quad [20]$$

Minkä jälkeen ratkaistaan yhtälöstä kulma  $\psi$ :

$$\psi = \arcsin\left(\frac{l_2 \sin(\theta_2)}{\sqrt{x^2 + y^2}}\right) \quad [21]$$

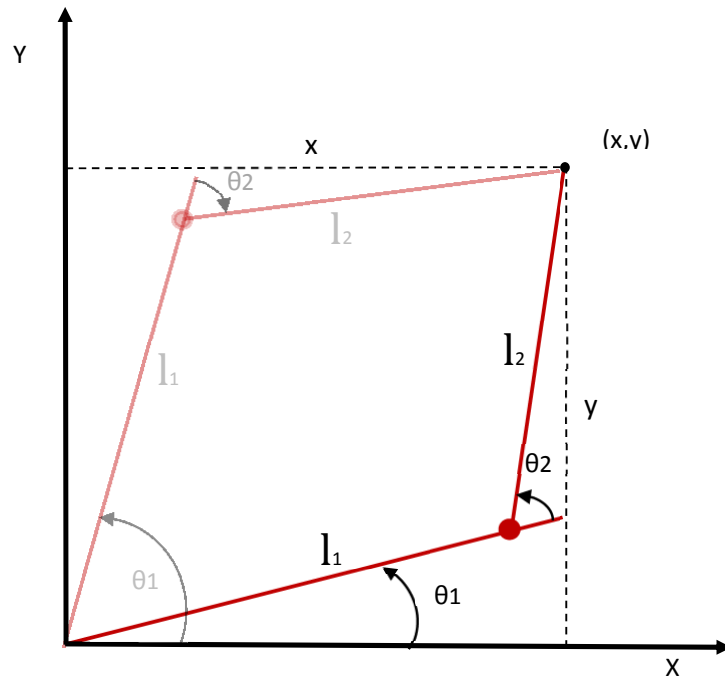
Kuvasta 52 huomataan lisäksi, että  $\psi$  voi olla joko positiivinen tai negatiivinen, riippuen kumpaa konfiguraatiota käytetään. Näin ollen kulma  $\theta_1$  saa arvot,

$$\theta_1 = \beta \pm \psi \quad [22]$$

Kulman  $\psi$  merkki on riippuvainen kulman  $\theta_2$  arvosta. Mikäli  $\theta_2 < 0$ , käytetään kulman  $\psi$  edessä positiivista merkkiä, kun taas  $\theta_2 > 0$  kulman merkki on negatiivinen. [Craig 2005]

### Algebrallinen ratkaisu

Kuvassa 53 on tasomanipulaattorin muuttujat, joita käytetään algebrallisen ratkaisun laskemisessa.



**Kuva 53.** Tasomanipulaattorin muuttujat. [Craig 2005]

Algebrallisen ratkaisun lähtökohtana on mekanismin siirtomatriisi. Kappaleen 2.2 perusteella tehty kokonaissiirtomatriisi, yhtälö 14 voidaan esittää myös seuraavassa muodossa:

$${}^0_2T = \begin{bmatrix} \cos \theta_{12} & -\sin \theta_{12} & 0 & x \\ \sin \theta_{12} & \cos \theta_{12} & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [23]$$

Kun tämä yhtälö merkataan yhtä suureksi yhtälön 14 kanssa, saadaan kaksi yhtälöä:

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \quad [24]$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \quad [25]$$

Molemmat yhtälöt korotetaan toiseen potenssiin ja summataan puolittain, sekä käytetään trigonometrisiä kaavoja:

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1l_2 \cos(\theta_2) \quad [26]$$

Lopputuloks on sama kuin geometrian avulla ratkaistu yhtälö 17.

Seuraavaksi ratkaistaan yhtälöstä  $24 \cos \theta_1$  ja saadaan:

$$\cos \theta_1 = \frac{x + \sin \theta_1 (l_2 \sin \theta_2)}{(l_1 + l_2 \cos \theta_2)} \quad [27]$$

Muokataan yhtälöä 25 trigonometrian kaavojen avulla ja asetetaan yhtälössä 27 saatu  $\cos \theta_1$ :n arvo yhtälöön:

$$y = \frac{x + \sin \theta_1 (l_2 \sin \theta_2)}{(l_1 + l_2 \cos \theta_2)} (l_2 \sin \theta_2) + \sin \theta_1 (l_1 + l_2 \cos \theta_2) \quad [28]$$

Yhtälön oikea puoli kerrotaan auki ja yhdistellään termit:

$$y = \frac{1}{(l_1 + l_2 \cos \theta_2)} \left( x l_2 \sin \theta_2 + \sin \theta_1 (l_1^2 + l_2^2 + 2l_1l_2 \cos \theta_2) \right) \quad [29]$$

Tämän jälkeen oikean puolen kosinilause  $(l_1^2 + l_2^2 + 2l_1l_2 \cos \theta_2)$  korvataan hypotenuusan neliöllä, eli termillä  $(x^2 + y^2)$ . Tämän jälkeen yhtälöstä ratkaistaan  $\theta_1$  ja lopullinen yhtälö näyttää seuraavalta:



$$\theta_1 = \arcsin\left(\frac{y(l_1 + l_2 \cos \theta_2) - x l_2 \sin \theta_2}{x^2 + y^2}\right) \quad [30]$$

Opetuslaitteen avulla voidaan todentaa laskennalliset arvot ja Excel- laskentataulun avulla voidaan tarkistaa suoran- ja käänteisen kinematiikan arvojen paikkansapitävyys.

Laskentataulukko liitteenä (Liite 1).

Taulukossa 2 on Fanuc M-16iB robotin D-H parametrit. Tästä taulukosta muodostetaan jokaiselle nivelelle siirtomatriisi. Nämä matriisit kerrotaan yhteen ja saadaan kokonaissiirtomatriisi. Tämä matriisi on huomattavan iso ja erittäin hankala käsin ratkaistavaksi. Idea on kuitenkin samankaltainen kuin tasomanipulaattorin esimerkki.

**Taulukko 2.** Fanuc M16iB Teollisuusrobotin D-H parametrit. [Fanuc 2012]

<i>joint</i>	$\theta_i$	$a_i$	$b_i$	$\alpha_i$	Min.	Max.
1	$\theta_1$	0.150	0.525	$-\pi/2$	-2.9671	2.9671
2	$\theta_2$	0.770	0.0	0.0	-2.1817	2.1817
3	$\theta_3$	0.10	0.0	$\pi/2$	-4.0143	4.0143
4	$\theta_4$	0.0	0.740	$-\pi/2$	-3.4907	3.4907
5	$\theta_5$	0.0	0.0	$\pi/2$	-2.4435	2.4435
6	$\theta_6$	0	0.10	0	-7.8540	7.8540
<i>unit</i>	<i>rad.</i>	<i>m</i>	<i>m</i>	<i>rad.</i>	<i>rad.</i>	<i>rad.</i>

### 4.3 Etäohjelmointiympäristö

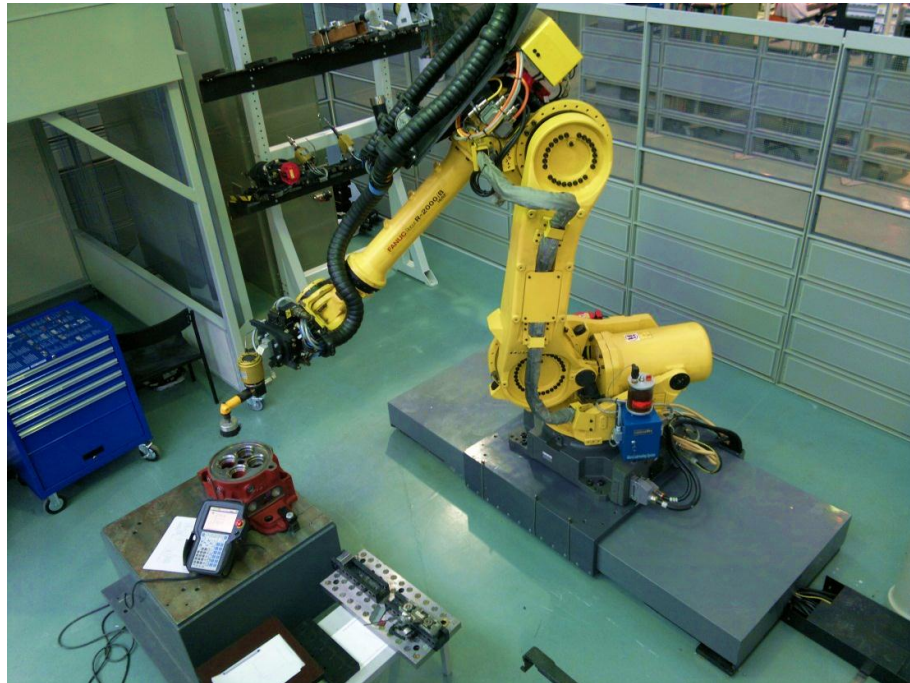
Etäohjelmoinnin tarkoituksena on luoda robottiratoja ja tehdä törmäystarkastelua, sekä optimoida työvaiheita. Tavoitteena on luoda ohjelmat erillisessä ympäristössä tuotantoa häiritsemättä ja simuloida toiminnot mahdollisimman kattavasti. Simuloinnin avulla havaitaan läheltä-piti ja törmäystilanteet, mahdolliset nivelraja- , nopeus- ja kiihtyvyyssylitykset, sekä singulariteetit. Valmis ohjelman tuotantoajo voidaan aloittaa nopeasti ilman pitkiä tuotannon keskeytyksiä ohjelmointityön vuoksi.

Törmäystarkastelu rajataan koskemaan tässä konseptissa työkalun ja työkappaleen välisiä kontakteja. Soluun valitaan robotti, mallinnetaan työkalu ja työkappale, sekä tarpeelliset oheislaitteet. Robotti löytyy avoimista kirjastoista, mutta muut osat joudutaan mahdollisesti mallintamaan itse, kuten robotin jalusta tai lineaarirata sekä työkalut. Ideana on mallintaa vain ne osat mitä radan luomisessa tarvitaan. Lisäksi solun kalibrointi tehdään robotilla määriteltujen työkalu- ja koordinaatistodatan avulla, jotka siirretään käsin mallin määrittelydataksi. Fastsimussa on paljon tehokkaita kalibrointityökaluja, mutta tässä vaiheessa niitä ei käytetä. Näin konsepti säilyy mahdollisimman yksinkertaisena ja helppokäyttöisenä. Oikealla robotilla olevat työkalunvaihtoproseduurit ja muut perusohjelmaan liittyvät osiot säilytetään. Robotille viedään ainoastaan ohjelma, jota kutsutaan robotin perusohjelmassa. Muodostettu ohjelma sisältää vain liikekäskyt, paikoiukset, mahdolliset I/O- komennot ja aliohjelmakutsut.

#### 4.3.1. Testiympäristön luonti ja verifiointi

Testiympäristöksi valikoitui Wärtsilän Technobothnia- laboratorioon sijoittama jäys-teenpoistosolu, kuva 54. Siihen sisältyy seuraavat osat:

- Lineaariradalla varustettu Fanuc R2000iB/165kg RJ3iC ohjauksella
- työkalut, servo- ja paineilmatyökaluja
- työkaluvaihtojärjestelmä ja työkaluteline (kymmenen paikkaa)
- suojaseinät, sisältäen siirrettävän koneistussuojan
- hiomapaperin vaihtoteline (Flap grinder- työkalun palvelu)



**Kuva 54.** Etäohjelmoinnin testiympäristö, ”Wärtsilä”- robotti Technobothniassa.

Tavoitteena oli mallintaa ja verifioida ”Wärtsilä”- jäysteenpoistorobotti valituilla työkaluilla. Aluksi täytyy tarkastella ja etsiä kriittiset tekijät ympäristön toiminnan kannalta. Tarkoituksena oli mallintaa vain robotin ratojen luontiin vaadittavat elementit: Robotti, työkalut, työkappale, robotin rata ja pöytä. Nämä osat tulee sijoittaa ja määrittää vastaamaan oikean robottisolun mittoja.

Etäohjelmointiympäristön luonti jakaantuu tässä konseptissa kahteen vaiheeseen, robotin valmisteluun ja simulointimallin määrittämiseen. Jos tarkoituksena on suunnitella uutta solua, niin silloin ei voida käyttää hyväksi olemassa olevia työkalu- ja koordinaatistomäärittelyjä. Tällöin solun valmistuttua täytyy solu ja malli kalibroida vastamaan toisiaan.

#### **4.3.2. Robottisolun valmistelu**

Lähtökohtana on toimiva ja kalibroitu robotti. Robotin sisäiseen kinematiikkaan ei kiinnitetä huomiota, luotetaan siihen, että nollatyökalun hallinta peruskoordinaatistossa on ohjaimella kunnossa. Ensimmäiseksi työalueelle kiinnitetään kalibroitipiikki, jonka avulla robotin työkalut määritetään käyttöohjeen mukaisesti. Tämän jälkeen kalibroidul-

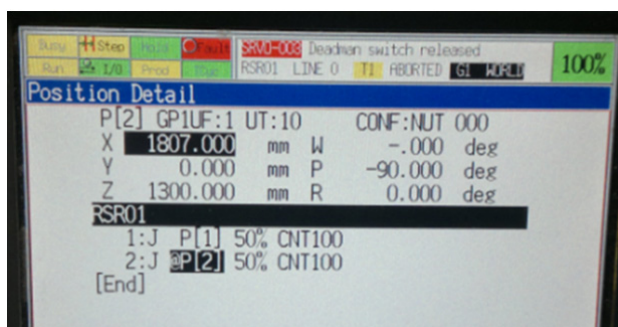
la työkalulla opetetaan käyttäjäkoordinaatisto työkappaleeseen tai kappaleen kiinnittimeen. Lisäksi valitaan etäohjelmointiympäristössä käytettävät työkalut, jotka on mallinnettava, ellei niitä löydy mallikirjastoista tai valmistajan internetsivuilta.

Robotille ohjelmoidaan nivelkoordinaatistossa oleva paikoituspiste, jossa nivelkulmat ovat nollakulmassa, eli robotti ajetaan 0-asentoon, kuva 55.



**Kuva 55.** Oikean robotin 0-asento.

Käyttäjäkoordinaatisto Uframe1 määritetään tässä tapauksessa vastaamaan robotin peruskoordinaatistoa. Työkalumäärittäyksessä valitaan jokin työkalu nollatyökaluksi, esimerkiksi UTool 10, jonka z-akseli asetetaan työkalulaippaa kohti ( $R_x=180\text{deg}$ ). Kuvasssa 56 näkyy paikoituspiste P2:n paikoitus peruskoordinaatistossa kun nivelet ovat nolla-asennossa.



**Kuva 56.** Oikean robotin P2 paikoitus 0-asennossa.

Kuvissa 56 ja 57 sama paikoitus oikealla robotilla ja mallin robotilla, arvot täsmäävät, ja voidaan varmistua että robottien asennot vastaavat toisiaan ja työkalu, sekä oheislaitteet voidaan lisätä vastaamaan oikean robotin paikoituksia. Tarkempi kalibrointi tehdään myöhemmissä vaiheissa.

#### 4.3.3. Simulointimallin luonti

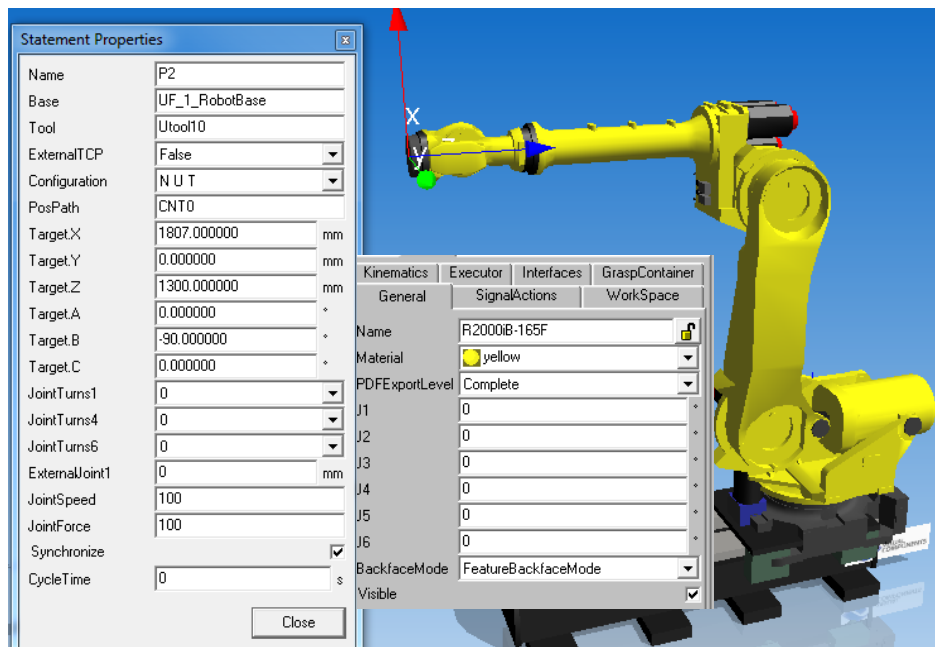
Etäohjelmointisolun luonnin työvaiheet ovat seuraavat:

1. robottityypin valinta
2. työkalun tuonti kirjastosta tai 3D-tiedostona
3. työkalun kiinnitys työkalulaippaan (robotilta työkaludata)
4. työkappaleen paikoitus työalueelle (robotilta koordinaatistodata)
5. radan luonti ja simulointi
6. ohjelmien siirron testaus molempiin suuntiin (malli>robotti>malli)
7. paikoitusten tarkkuuden testaus, verifiointi.

#### Robotti soluun

Ensimmäisenä vaiheena on tuoda robotin jalusta tai lineaarirata soluun. Sen voi tuoda 3D-mallina tai valita kirjastosta ja paikoittaa soluun. Jos robotti on asennettu lineaariradalle, tulee käyttää kirjaston ratamallia ja määrittää sen parametrit vastaamaan oikeaa rataa. Robotti valitaan kirjastosta ja paikoitetaan haluttuun paikkaan maailmankoordinaatistossa radan tai jalustan päälle. Mallin robotti paikoitetaan 0-asentoon ja tarkastetaan 0-työkalulla mallin ja oikean robotin perusasento kappaleen 4.3.2 mukaisesti

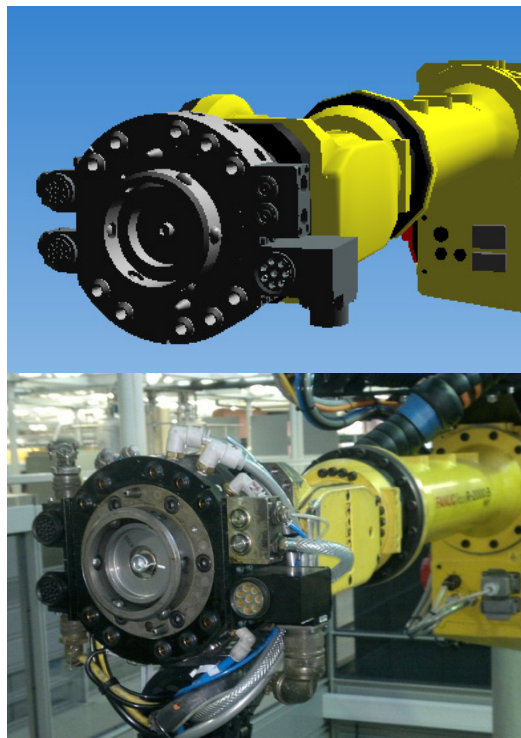
Fastsimussa on kahdenlaisia 3D-malleja, pintamalleja ja toiminnallisia malleja. Pintamalli on pelkkä kolmiulotteinen kuva kappaleesta. Toiminnalliset mallit sisältävät simuloinnissa käytettäviä toimintoja, kuten robotin kinematiikka, työkalujen kiinnityspisteet ja tarraimien liikkeet. Nämä toiminnallisuudet voidaan luoda malleihin esimerkiksi Visual Componentsin 3D-Create ohjelmalla. Fastsimussa ei ole mahdollista luoda toimintoja malleihin, vaan siinä käytetään valmiita komponentteja, joissa kinematiikka on valmiina. Esimerkiksi mallikirjaston työkaluissa on toiminnallisuus, jonka avulla työkalut ovat helppo kiinnittää työkalulaippaan Fastsimun plug and play- toiminnolla. Tämän työn lähtökohtana käytetään myös malleja, joissa ei ole toiminnallisuutta.



*Kuva 57. Robotin 0-asento, kuvassa myös nivelien asentotieto.*

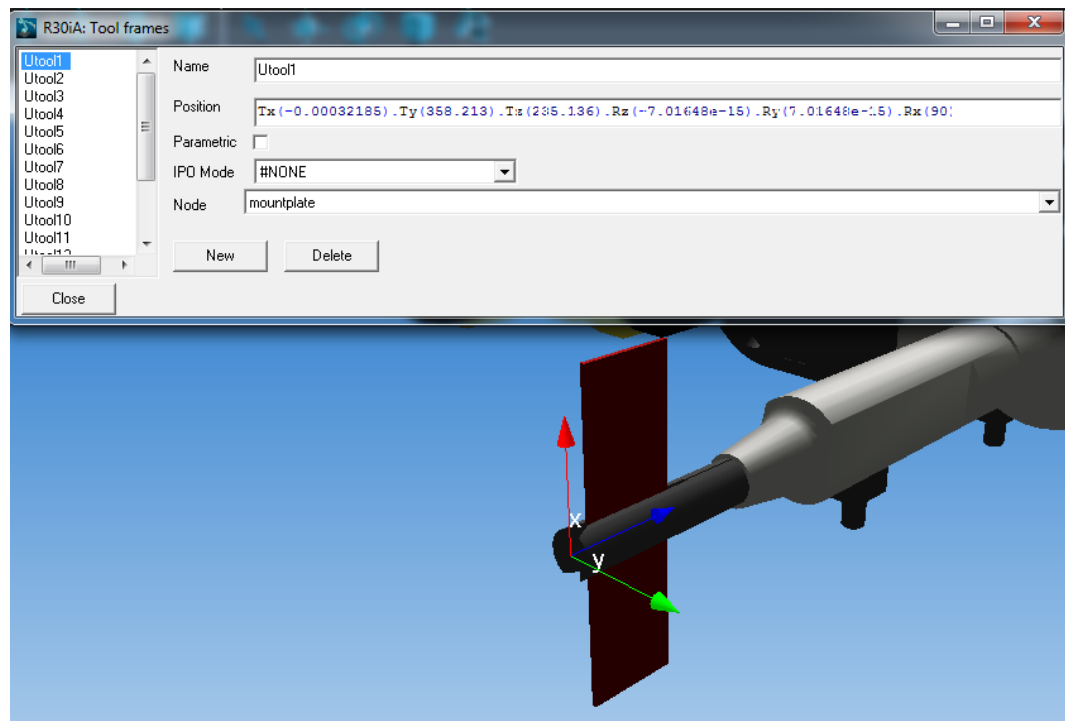
### Työkalujen lisäys

Työkalujen tai mahdollisen adapterin lisäys tehdään robotin 0-asennossa, jolloin työkalun asento ja kiinnitys saadaan myös visuaalisesti tarkastettua.



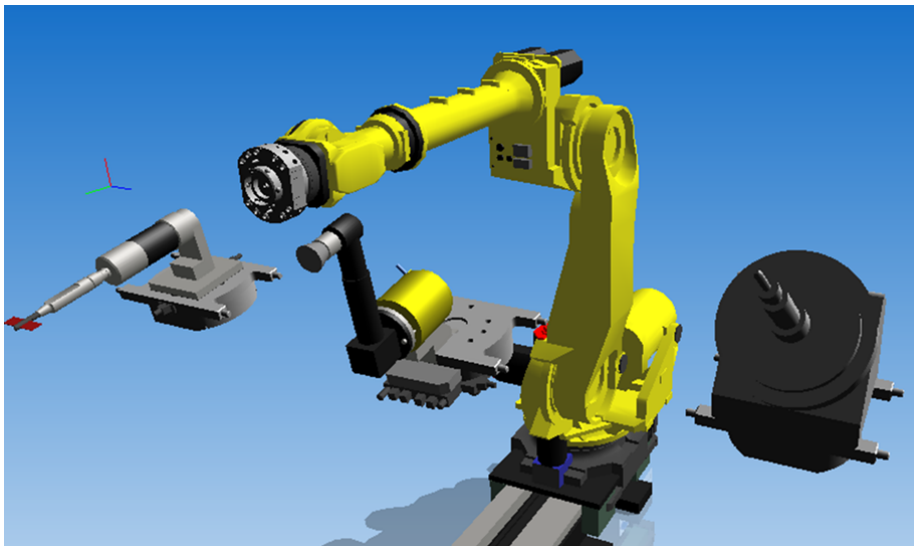
*Kuva 58. Työkaluadapterin kiinnitys ja asennon tarkastus 0-asennossa.*

Työkaluadapteri kiinnitetään ensimmäisenä robotin työkalulaippaan. Mallit haetaan joko Fastsimun kirjastosta tai vaihtojärjestelmän valmistajan internetsivuilta. Jos malleja ei ole, täytyy ne mallintaa erillisellä 3D-ohjelmalla ja siirtää mallit soluun. Kuvassa 58 on molemmat robotit nolla-asennossa ja adapteri kiinnitetty. Tämän jälkeen voidaan varsinainen työkalu kiinnittää adapteriin. Työkalun määrittys tehdään oikealla robotilla käyttöohjeen mukaisesti kuuden pisteen menetelmän avulla. Tämän jälkeen mallin työkaluun asetetaan robotin työkaludata, kuva 59. Jos koordinaatisto ei ole oikeassa kohdassa, 3D-mallia siirretään siten, että koordinaatisto on oikeassa paikassa ja asennossa. Kuvassa 59 on Flap grinder- työkalun työkalukoordinaatisto.



**Kuva 59.** Mallin työkalukoordinaatiston määrittäminen.

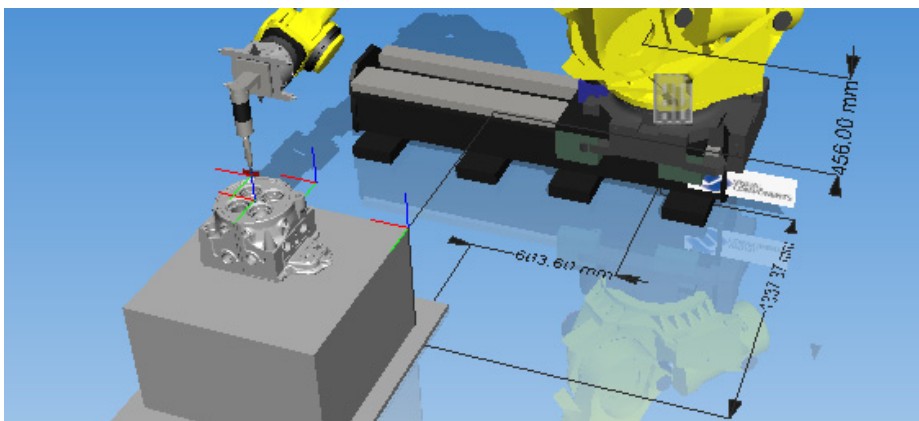
Työkalut liitetään ja irrotetaan tarpeen mukaan parent-komennolla työkaluadapterista. Kuvassa 60 on työkalut irrotettuina. Tässä esimerkissä työkalut kiinnitetään manuaalisesti, eikä siis simuloita työkaluvaihtoa. Työkaluvaihto-ohjelma on jo robotilla, joten sen simulointi ei tuo tässä vaiheessa uutta informaatiota. Toisaalta työkierron sisältäessä useita työkaluvaihtoja, tahtiaikojen selvittämiseksi myös työkaluvaihdot tulisi simuloida.



*Kuva 60. Työkalut "telineissään".*

### **Työkappaleen paikoitus**

Kun työkalut ovat määriteltynä, voidaan työkappale ja tarvittavat pöydät ja kuljettimet lisätä soluun. 3D-mallit noudetaan mallikirjastosta tai tuodaan omat mallit soluun esimerkiksi STEP- formaatissa, kuten työkalutkin. Tarkkaan paikoitukseen käytetään oikealla robotilla määritetyn käyttäjäkoordinaatiston dataa. Robotilla määritetään koordinaatisto aiemmin määritetyllä työkalulla ja tämä paikoitusdata siirretään mallin koordinaatiston paikoitusdataksi.



*Kuva 61. Pöytä paikoitettuna robotin peruskoordinaatistoon radan 0-asennossa.*

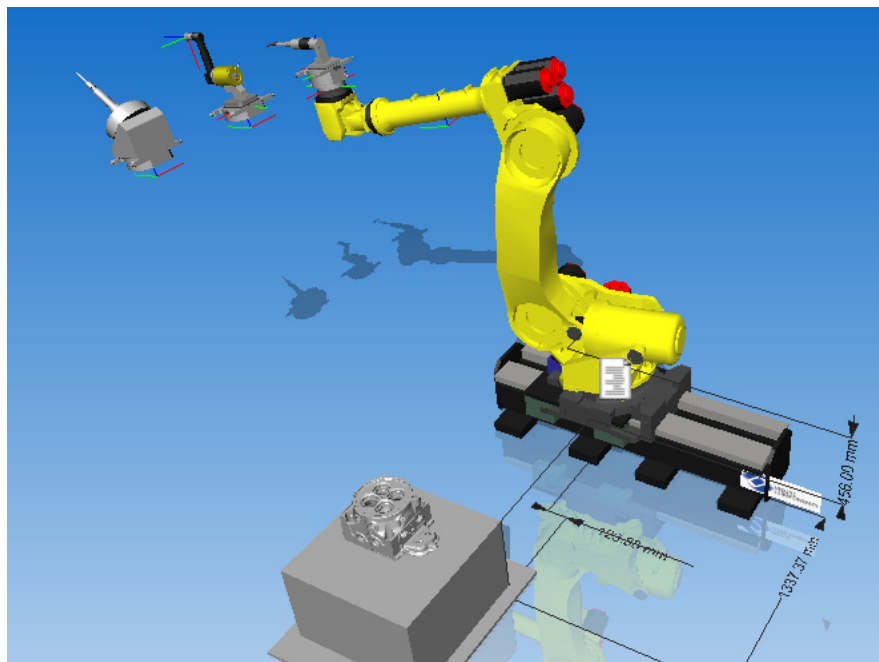


Kuvassa 61 on pöydän paikoitukseen liittyvät mitat. Oikealla robotilla määritellyt käyttäjäkoordinaatiston arvot syötetään mallin koordinaatiston arvoiksi ja pöytä siirretään ko. koordinaatiston mukaiseen paikkaan.

Työkappale voidaan tuoda kalibroidulle pöydälle, jolloin täytyy huolehtia että paikka on sama mallissa ja oikeassa solussa. Jos koordinaatisto sijoitetaan työkappaleeseen, täytyy tarkastaa robotin mittatyökaluna käytettävän työkalun ulottuvuus ja pääsy sopiviin mittapisteesiin.

Soluun lisätään mallikirjastosta lineaarirata, jonka päälle robotti paikoitetaan. Lineaariradan mallin parametrit täytyy asettaa niin, että malli vastaa oikean radan liikeratoja. Lopuksi täytyy varmistaa mallin peruskoordinaatiston ja käyttäjäkoordinaatistojen vastaavuus oikeaan robottiin.

Kuvassa 62 on testiympäristöön valitut työkalut ja työkappale. Testiympäristön työkappaleena on Wärtsilän W20- moottorin sylinterinkansi. Se on oikea tuotantokappale ja tarkoituksena on kokeilla konseptia myöhemmin myös tuotannossa. Tarkoitus on tehdä kolmella eri työkalulla erilaisia työkiertoja ja testata ratojen siirtoa ja toimivuutta testisolussa.



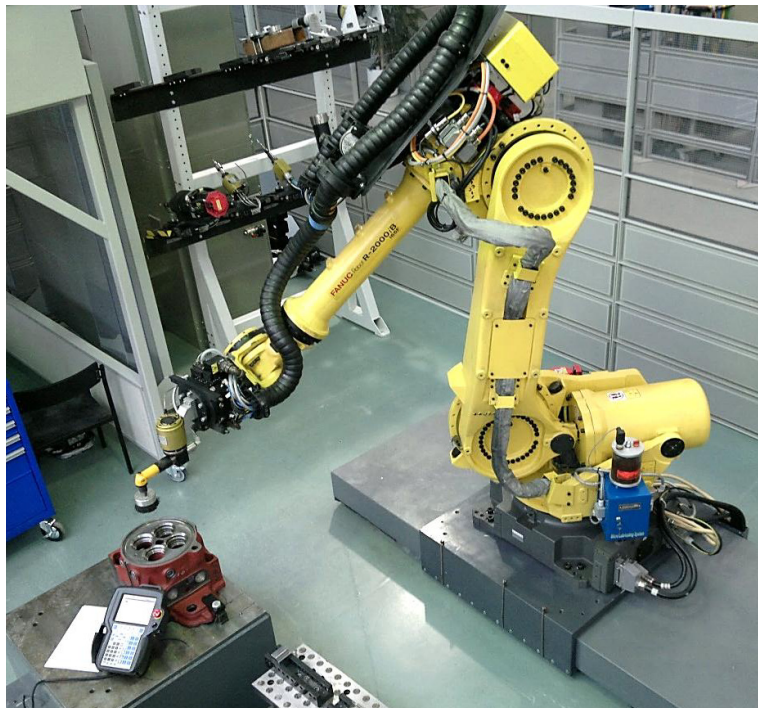
**Kuva 62.** Testiympäristö valittuine työkaluineen.

Kuvassa 63 on malli ja oikea robotti vierekkäin, koneviilatyökalu ja sylinterinkansi. Kiinnittimet ja kannen asento eivät vastaa tuotantotilannetta, mutta tarkoituksena onkin tässä vaiheessa luoda mallin avulla liikeratoja oikealla robotilla toteutettavaksi.



**Kuva 63.** Simulointimalli ja oikea robotti.

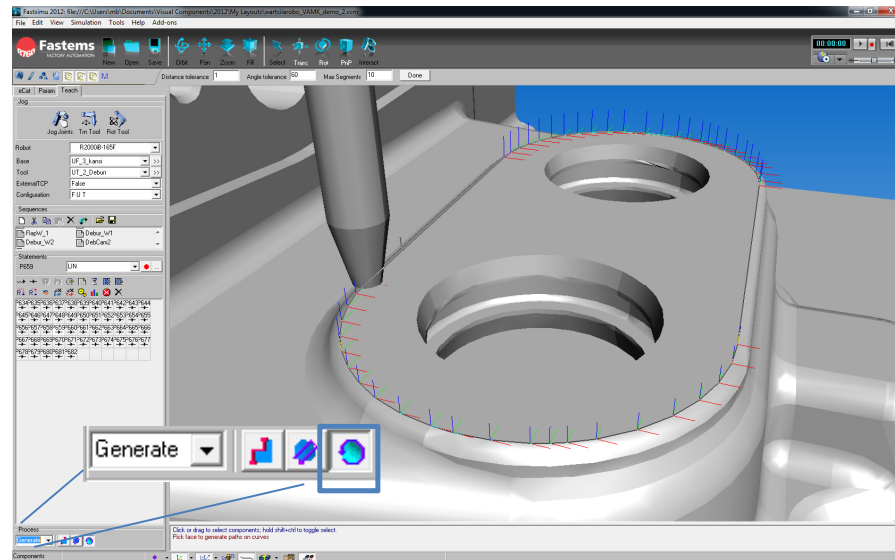
Kun työkalut ja muut tarpeelliset oheislaitteet ovat mallinnettu ja paikoitettu oikein, voidaan aloittaa varsinaisten ratojen luonti ja ohjelmien siirto robotille. Kuvassa 64 testisolu harjatyökalu kiinnitettynä.



**Kuva 64.** Testisolu.

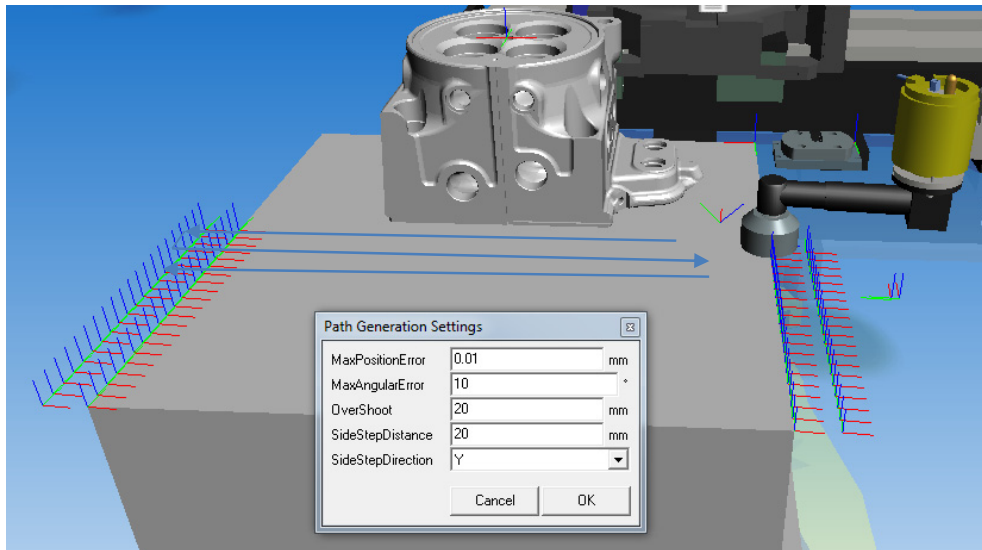
## Radan luonti ja simulointi

Fastsimussa rata voidaan luoda muutamalla eri tavalla. Lähtökohta on aina kuitenkin sama kuin oikeankin robotin ohjelmoinnin aloitus, eli valitaan käytettävä työkalu ja koordinaatisto aktiiviseksi. Rata voidaan muodostaa paikoituspistein, kuten oikeallakin robotilla liikuttamalla robottia haluttuihin paikoituksiin ja määrittelemällä liike ko. paikoitukseen. Toinen tapa on käyttää työkappaleen piirteitä apuna ja muodostaa rata valittujen käyrien ja tasojen avulla. Kuvassa 65 on esitetty työkappaleen erään reunakäyrän avulla muodostettu rata. Kuvassa myös suurennos käytetyn menetelmän valintapainike.



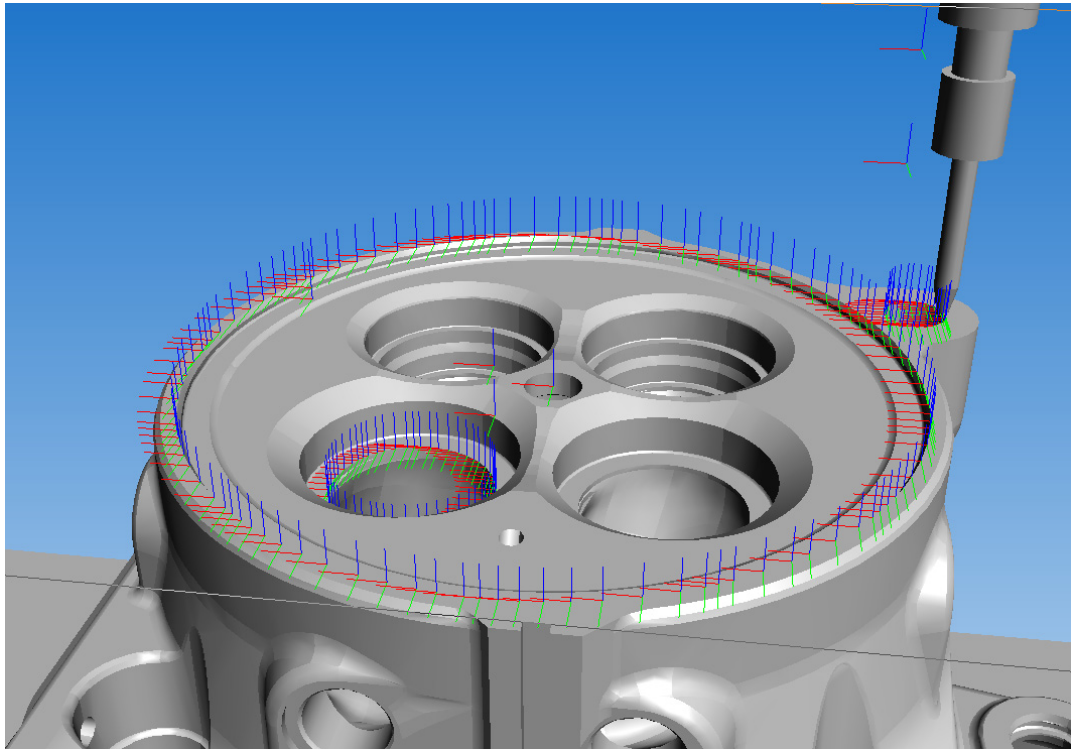
*Kuva 65. Radan luonti valitulle reunakäyrälle.*

Tasojen avulla voidaan määritellä edestakainen pyyhkäisy esimerkiksi harjatyökaluille. Radan parametrina voi antaa sivuttaissiirron ja ylituksen arvot. Kuvassa 66 on tason avulla muodostettu harjaustyökierto ja pyyhkäisyparametrit. Sininen nuoli osoittaa työkalun liikeradan.



**Kuva 65.** Radan luonti tason pyyhkäisyyn.

Ratojen luonnin yhteydessä voidaan valita myös muita parametreja, kuten työkalun kulma ja paikoitustoleranssi. Fastsimussa on käytössä lineaariliike ja nivelliike. Kaaret toteutetaan usealla pisteellä ja isolla CNT- arvolla. CNT- arvo (Continuous) voidaan antaa liikekäskyyn parametrina välillä FINE, 0...100. FINE tarkoittaa että TKP kulkee tarkasti paikoituspisteeseen, pysähtyy ja jatkaa sitten seuraavaan paikoitukseen. CNT 0 menee paikoituspisteeseen, mutta ei pysähdy. CNT- arvolla 100 TKP ei mene lainkaan ohjelmoidun pisteen kautta, vaan alkaa lähestyä seuraavaa paikoitusta ennen kuin ohjelmoitu piste toteutuu. Muilla arvoilla TKP oikaisee 0 ja 100 väliltä. Oikaisuetäisyys riippuu myös liikenopeudesta. Sopivalla CNT- arvolla liikerata on kaareva ja silloin ympyränkaaret voidaan toteuttaa lineaariliikkeenä. Kuvassa 66 on generoitu kolme eri rataa kannen yläosan muotoihin. Kuten kuvasta näkyy, paikoituspisteitä syntyy ympyränkaarille useita. Paikoituspisteiden määrä riippuu valitusta tarkkuudesta, eikä suurikan määrä ole ongelma, koska Fastsimussa on monipuoliset editointityökalut suurienkin pisteryhmien editointiin kerralla. Jos tarkoituksena on muokata ohjelmaa käsiohjaimella, kannattaa minimoida paikoituspisteiden määrä.



**Kuva 66.** Sylinterikannen yläpinnalle generoituja paikoituspisteistä muodostuvia ratoja.

Paikoituspisteiden editointi on Fastsimussa tehokasta. Paikoituksia voidaan siirtää ja kääntää, kaikkia kerrallaan tai vain valittuja, valitun koordinaatiston suunnassa tai valitun referenssipaikoinnin suunnassa.

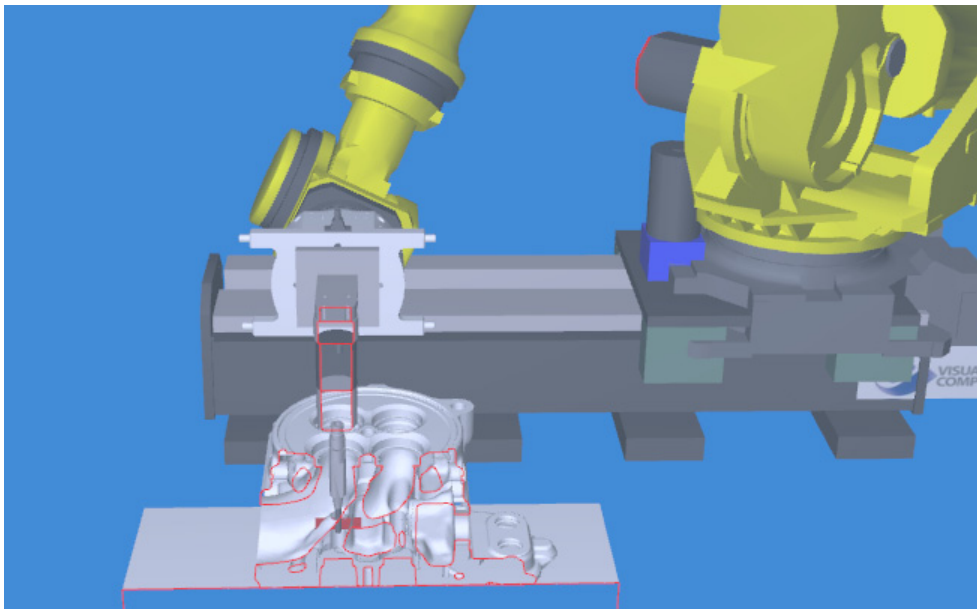
Kun varsinainen rata on valmis, lisätään siihen lähestymis- ja poistumispisteet. Jos työkaluvaihtoja ei simuloida, kannattaa lisätä ohjelman alkuun ja loppuun liikekäskyt siihen paikoitukseen, mistä työkaluvaihto-ohjelma alkaa ja minne päättyy. Tämän menettelyn avulla simuloinnin lähtökohtana on sama nivelkonfiguraatio kuin oikeassakin robotissa. Jos konfiguraatiot ovat erilaiset, robotti tekee turhia kiertoliikkeitä vaihtaessaan niveliensä asentoja. Kiertoliikkeen aikana työkalulle menevät kaapeloinnit saattavat kiertyä liiaksi ja rikkoutua.

### **Simulointi**

Vielä ennen ohjelman kääntöä robotin ymmärtämään muotoon eli postprosessointia, liikkeet täytyy simuloida ja tarkastaa niiden sopivuus. Tavoitteena on estää törmäykset, minimoida turhat liikkeet ja käännöt, sekä asettaa prosessiparametrit sopiviksi. Jokaisen radan alkuun kannattaa lisätä nivelliike, koska se määrää käytettävän konfiguraation ja

lineaariliike toteuttaa aina olemassa olevaa konfiguraatiota. Näin menetellen käyttäjä voi käyttää aina sopivinta konfiguraatioita ja turhat käännöt voidaan minimoida. Lisäksi nopeudet ja CNT- arvot säädetään vastaamaan sopivaa työstöprosessia.

Fastsimun avulla simulointi saadaan myös PDF-muotoiseksi videoksi. Videossa on paljon käyttökelpoisia ominaisuuksia, kuten zoomaus, kuvakulman vaihto, leikkaukset ja valaistuksen säätely. Videoiden avulla simulaatiot voidaan jakaa käytännössä kaikkien kanssa. PDF- tiedostot ovat tehokkaasti pakattu, joten tiedostokoko on sopiva jopa sähköpostin liitteeksi. PDF- lukijaohjelmat ovat yleisimmissä käyttöjärjestelmissä toimivia ilmaisohjelmia.



*Kuva 76. Kuvakaappaus simulointivideosta ja leikkauksesta.*

### Postprosessointi

Robotille viedään vain työstöohjelma ja paikoitukset. Työkalu- ja koordinaatistonumeroinnit on oltava samoja kuin reaalirobotilla. Robotilla käytetään olemassa olevaa ohjelmaa, jossa kutsutaan Fastsimulla tehtyä liikeradat sisältävää ohjelmaa. Tässä esimerkiksi robotin ohjelma, jossa haetaan ja määritetään oikeat työkalu- ja käyttäjäkoordinaattit. Varsinainen rata sijaitsee FLAP2- ohjelmassa.

```
/MN
1:  CALL TK_VAIH2(9) ;
2:  UFRAME_NUM=3 ;
3:  UTOOL_NUM=3 ;
4:  CALL FLAP2      ;
/POS
```

/END

FLAP2 näyttää seuraavanlaiselta, siinä on vain liikeradat ja työkalun ohjauskäskyt. Ohjelman lopussa paikoituksien määrittelyt.

```
/MN
1:J P[1: P477] 50% CNT0 ;
2:J P[2: P19] 50% CNT60 ;
3:L P[3: P473] 500mm/sec CNT60 ;
4:L P[4: P26] 500mm/sec CNT60 ;
5: DO[449: S21]= ON ;
6:L P[5: P474] 100mm/sec CNT60 ;
7: WAIT 1.00(sec) ;
8:L P[6: P475] 100mm/sec CNT60 ;
9: DO[449: S22]= OFF ;
.
.
.
19:L P[13: P483] 500mm/sec CNT60 ;
20:J P[14: P471] 50% CNT0 ;
/POS

P[1]{
GP1:
UF : 3, UT : 3, CONFIG : 'F U T, 0, 0, 0',
X =668.92 mm, Y =-1857.14 mm, Z =717.35 mm,
W =-89.70 deg, P =-1.03 deg, R =84.71 deg,
E1 =400.00 mm
};
```

Fastsimulla ohjelma käännetään tekstimuotoiseksi, minkä jälkeen se siirretään robotille. Jos robotin ohjaimesta puuttuu tekstimuotoisten ohjelmien latausmahdollisuus, pitää ohjelma kääntää ensin binäärimuotoon Fanucin ohjelmistolla.

### Verifiointi

Lopuksi ohjelmat ja radat ajetaan hitaalla nopeudella oikealla robotilla ja samalla tarkkaillaan paikoitustarkkuuksien riittävyys. Tässä työssä päähuomio on menetelmän käyttöönotossa ja sen opettamisessa, joten paikoitustarkkuuden mittaus rajataan työn ulkopuolelle. Testiympäristössä suoritettut koeajot olivat silmämääräisesti mitattuna paikoitustarkkuudeltaan riittävän tarkkoja jäysteenpoistosovelluksiin. Poikkeamat olivat vain yhden millimetrin luokkaa. Virheitä tulisi tarkastella aina tapauskohtaisesti, koska solujen virhelähteet ja kokoluokka vaihtelevat eri sovelluksissa. Jäysteenpoistossa käytetään lähes aina joustavia menetelmiä, joten etäohjelmoinnin paikoitusvirheet kompensoituvat prosessissa helposti. Fastsimun avulla voidaan tehdä nopeasti muutoksia isoonkin paikoitusjoukkoon ja sitä kautta parantaa ohjelmoitua rataa.



Valmis ympäristö esiteltiin myös Wärtsilän robotiikan vastuuhenkilöille. Esittelytilaisuudessa tehtiin myös muutama ohjelma simulointiympäristössä ja siirrettiin ne robottle. Tilaisuudessa esiteltiin ainoastaan Fastsimun pääpiirteet ja tavoitteena olikin näyttää konseptin toiminta ja herättää ideoita jatkoa ajatellen. Palaute oli erittäin positiivista ja ideoita etäohjelmoinnin käytöstä syntyi jo ensi hetkistä.

## 4.4 Harjoitukset

Tavoitteena on luoda harjoituksia, joiden avulla voidaan kokeilla tärkeimpiä robotin käyttöön liittyviä toimintoja. Harjoitukset on jaettu robotin peruskäyttöharjoituksiin ja etäohjelmointiharjoituksiin. Harjoitukset testataan Technobothnian pienemmällä Fanuc-robotilla M-6iB:lla. Samalla tehdään ohjeet oppimisympäristön harjoituksien suorittamiseksi. Tässä tehdään käytännössä toinenkin etäohjelmointiympäristö, jonka avulla varmistetaan ohjeet ja menetelmät.

### 4.4.1. Peruskäyttöharjoitukset

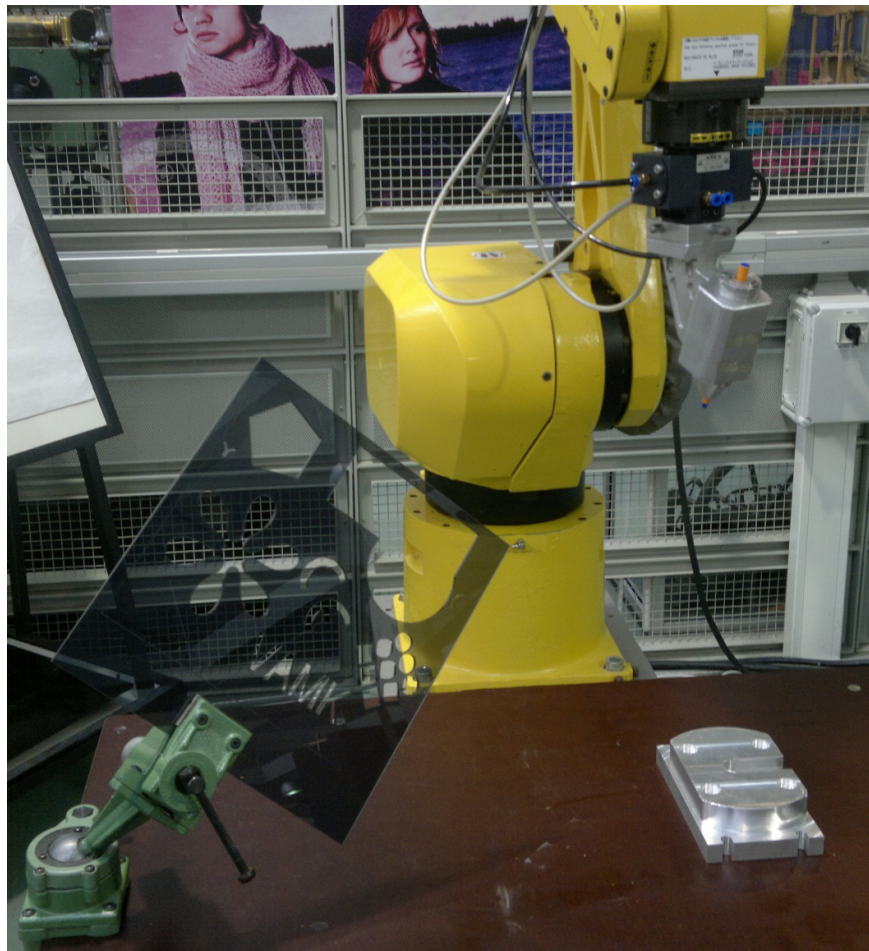
Robotin peruskäyttöharjoitukset sisältävät harjoituksia robotin liikuttamiseksi käsiohjaimella ja liikeohjelman tekoon. Liitteessä 4 on peruskäyttöharjoituksien tehtävannot. Sisältö on seuraavan listan mukainen:

- robotin liikuttelu käsiohjaimella eri koordinaatistoissa
- käsiohjaimen muiden toimintojen läpikäynti (Select, Edit, sallintalaite)
- työkalun ja käyttäjäkoordinaatiston määrittäminen (ToolFrame, UserFrame)
- paikoitusten opetus (points)
- liikekäskyjen ohjelmointi (Linear, Circular)
- aliohjelmien käyttö (CALL)
- ehto- ja hyppylausekkeet (IF, JMP)
- ohjelman testaus, tallennus ja lataus
- rekisterien käyttö (R[5]=5)
- tulojen ja lähtöjen ohjaus (DO[14]=off).



#### 4.4.2. Etäohjelmointiharjoitukset

Etäohjelmointiharjoitusten työkappaleina käytetään kahta erilaista työkappaletta, muotoleikattua muovilevyä ja alumiinista kannatinta. Harjoitustöiden tehtävänannot ovat liitteessä 5. Soluun tuotavista osista on valmiina 3D- mallit STEP- formatissa. Työkaluna käytetään samaa kynätyökalua kuin perusharjoituksissakin. Kuvassa 77 on etäohjelmointiharjoitusten komponentit: robotti, jalusta, kynätyökalu, pöytä, kannatin ja harjoituslevy.



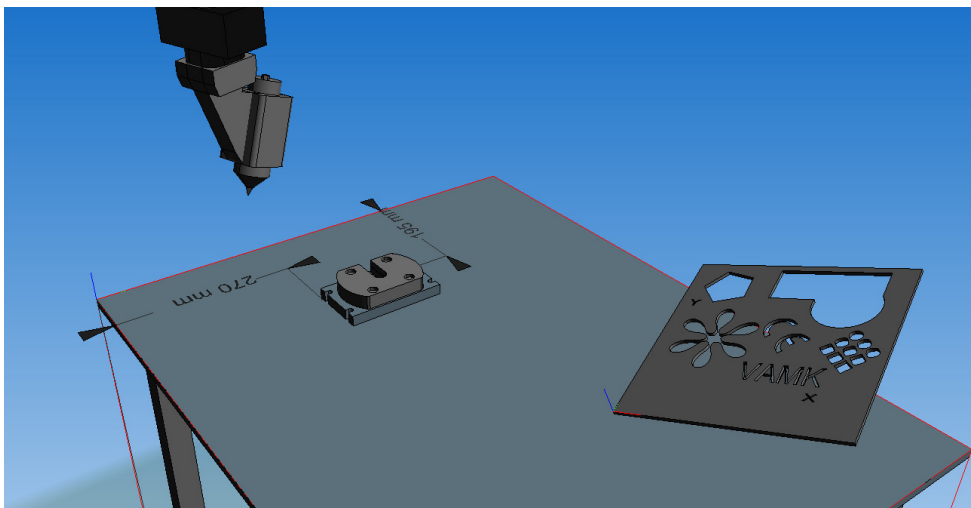
*Kuva 77. Etäohjelmointiharjoitusympäristön osat.*

Tarkoituksena on tuoda valmiit 3D- mallit soluun ja paikoittaa ne robotin koordinaattistojen avulla paikoilleen. Kaikki alkaa robotin ja jalustan tuomisesta malliin, jonka jälkeen työkaluadapteri ja työkalu kiinnitetään. Tämän jälkeen tuodaan pöytä ja työkappaleet mallin työalueelle. Oikeassa solussa kannatin asetetaan pöydälle ja harjoituslevy ruuvipuristimeen. Työpöydän ja harjoituslevyn kulmiin opetetaan käyttäjäkoordinaatis-

tot oikealla robotilla ja näiden arvojen avulla työpöytä ja harjoituslevy paikoitetaan so-  
luun.

Harjoituksessa käydään läpi kaksi erilaista toimintatapaa. Ensimmäinen tapa käyttää pöydän koordinaatistoa ja kannatin asetetaan pöydälle tarkasti samaan paikkaan mallis-  
sa ja oikealla robotilla. Toisessa tavassa käyttäjäkoordinaatisto opetetaan työkappalee-  
seen, tässä tapauksessa harjoituslevyyn. Levy paikoitetaan malliin opetetun koordinaa-  
tiston mukaisesti.

Kuvassa 78 on malliin paikoitetut työkappaleet ja pöytä. Ruuvipuristinta ei ole mal-  
linnettu vielä. Se mallinnetaan seuraaviin harjoituksiin törmäystarkastelujen suorittamis-  
ta varten.



**Kuva 78.** Etäohjelmointiharjoitusympäristön osat.

Perusidea on, että ohjaaja asettaa työkappaleet oikealla robotilla sattumanvaraisesti paikoilleen. Tämän jälkeen opiskelija määrittelee tarvittavat koordinaatistot oikealla robotilla, sekä siirtää mallin osat oikeille paikoilleen. Seuraavaksi ohjaaja määrittelee työkierrat valituille piirteille ja opiskelija tekee ohjelmat, simuloi ja ajaa ohjelman oikealla robotilla. Työkappaleiden paikoitustieto ja työkiertomäärittelyt voivat olla mukana työohjeessakin. Eri ryhmille on helppo antaa erilaiset paikoitukset ja näin vältetään ohjelmien kopiointia.

Harjoituksesta voi tehdä myös etäopiskeluversion, missä oppilas palauttaa kommentoidun ohjelmalistauksen lisäksi simulaation PDF- formaatissa. Nämä kehitetyt harjoitukset ovat aivan perusharjoituksia, mutta näiden avulla peruselementit tulevat tutuiksi. Jatko-opiskelua varten on todella paljon mahdollisuuksia laatia hyvinkin vaativia harjoituksia.

Harjoitustyöhön valmistettu levy suunniteltiin ja valmistettiin laserkaivertimella. Siinä olevan ohjelmiston avulla kaikki mikä on mahdollista piirtää tai kirjoittaa PC:n ohjelmilla, saadaan laserin avulla leikattua tai merkattua. Lisäksi luodut kuvat voidaan tallentaa IGES- formaattiin, mikä on yksi paljon käytetyistä CAD- järjestelmien siirtoformaateista. Tässä tapauksessa levyn IGES- muotoinen 2D- tiedosto avattiin NX- ohjelmassa ja siihen lisättiin levyn paksuus pursottamalla. Sen jälkeen se tallennettiin STEP- formaattiin ja vietiin Fastsimun etäohjelmointiympäristöön. Kaikki muutkin mallinnetut osat luotiin NX- ohjelmalla, eikä tiedostojen siirrossa ollut ongelmia.

#### **4.5 Robotin ja etäohjelmoinnin ohjeet**

Koska robotin ohjekirja on erittäin kattava, täytyi ohjeet tiivistää sellaiseen laajuuteen, minkä avulla tarvittavat harjoitteet saadaan tehdyksi. Käyttöohjeet robotille LIITE2 ja etäohjelmointiohjelmistolle LIITE3. Fastsimulle on olemassa myös valmistajan ohjeistusta, ne käsittelevät vain ohjelman toimintoja, ei niinkään uuden solun luomiseen liittyviä menetelmiä. Liitteen Fastsimu- ohjeet ovat tehty ainoastaan siihen laajuuteen, että niiden avulla pystyy tekemään perusharjoitukset.

## 5 YHTEENVETO JA PÄÄTELMÄT

Työssä selvitettiin mitä robotisoitu jäysteenpoisto sisältää ja mitä tulee osata, että jäysteenpoistosolun etäohjelmointiympäristö voidaan toteuttaa. Tehokas robottisoveltaminen etäohjelmointiympäristöineen vaatii enemmän osaamista kuin perinteinen robotin ohjelmointi käsiohjaimella. Kehitetty opetuslaitteisto havainnollisine esimerkkeineen vahvistaa koordinaatioajattelua, mikä on robotiikan perusta. Harjoitusten ja ohjeiden avulla voidaan tehdä ohjelmia käsiohjaimella, sekä luoda toimiva etäohjelmointiympäristö. Harjoitukset sisältävät ainoastaan perustoimintojen läpikäynnin, mutta antavat vankan pohjan sekä edellytykset jatko-opinnoille. Käytännön harjoitukset kehitetään jatkossa kaikkiin robottimerkkeihin sopiviksi ja erilaisien sovelluksien opetteluun. Oppilaitoksissa voidaan luoda oma ympäristönsä, ja yrityksissä voidaan käyttää olemassa olevaa laitteistoa harjoitteluvälineenä.

Robottisoveltamisessa ei riitä pelkästään ratojen luonti, vaan on ymmärrettävä lisäksi prosessi ja siinä käytettävät menetelmät sekä työkalut. On myös huomioitava että robottien sovellusohjelmat sisältävät paljon muutakin varsinaisien liikeratojen lisäksi. Kyseessä on monimutkainen ympäristö ja työ rajautui koskemaan etäohjelmoinnilla toteutettua radan ohjelmointia ja robotin liikeohjauksen mekanismeja. Tämä perusta on oltava kunnossa robotin ohjelmoimiseksi, varsinkin etäohjelmoinnissa. Työssä perehdyttiin jäysteenpoiston robotisointiin, ja sieltä nousikin esiin useita huomioita. Jäysteenpoistossa käytetään lähes aina joustoelementtejä, joita voivat olla työkalut, pitimet tai itse robotti voimaohjauksella. Jäysteenpoisto on viimeisiä käsityövaiheita konepajoissa, joten sen automatisointi olisi tärkeä askel kohti automaattista pienen eräkoon tuotantoa. Etäohjelmoinnin avulla robotit voidaan simuloida ja ohjelmoida etukäteen vaihtelevassa tuotannossa, sekä kehittää olemassa olevia ohjelmia keskeyttämättä tuotantoa. Usein kehitystyö jää kesken ja tyydytään välttävästi toimivaan järjestelmään. Jäysteenpoiston suunnittelussa tulisi huomioida myös jäysteen syntymekanismi, ts. voiko siitä tehdä helpommin poistettavan tai estää sen syntyminen kokonaan.

Fastsimun toiminnoista tässä työssä on käytetty ainoastaan murto-osa, vain yksinkertaisen esimerkkisolun luontiin vaadittavat työkalut. Kalibrointityökaluja ei käytetty lainkaan, vaan kalibrointi toteutettiin siirtämällä robotilla mitattu data malliin. Valittu tapa sopii hyvin koordinaatistojärjestelmien opetukseen, mutta jatkossa kalibrointityökalutkin on hyvä ottaa mukaan harjoituksiin. Liikenopeuksien vastaavuuden tarkastelu jätettiin myös työn ulkopuolelle. Joitain pieniä ohjelmistovirheitä tuli myös esille, mutta ne korjattiin työn edetessä.

Fastsimun peruskäytön opettelu onnistui hyvin pienellä perehdytyksellä. Ohjelmisto on hyvä opetuskäytössä, koska se on merkkiriippumaton ja selkeä. Simulointimalli voidaan toteuttaa hyvin yksinkertaisena, jolloin perusasiat pysyvät esillä. Jatkossa taas monipuoliset toiminnot voidaan ottaa käyttöön ja ohjelmiston todellinen tehokkuus tulee esiin. Opetuskäyttöä ajatellen olisi ensiarvoisen tärkeää, että opiskelijat voisivat käyttää ohjelmistoa myös koulun ulkopuolelta kelluvan lisenssin ja etäyhteyden avulla. Uusien ohjelmistojen oppiminen vaatii paljon harjoittelua, eikä oppilailla ole aina mahdollisuutta lähteä koululle. Vaasan ammattikorkeakoulun konetekniikan opetuksessa on suuntauduttu vahvasti 3D-tuotesuunnitteluun. Etäohjelmointiympäristön vaatimien 3D-mallien luominen syntyisi helposti muiden opintojaksojen sisällä. Tämä tuo hyviä mahdollisuuksia opintojaksojen kokoamiseen mielekkäiksi kokonaisuuksiksi. Mielenkiintoinen mahdollisuus huomattiin laserkaivertimen käytön aikana. Robottiohjelma voidaan luoda helposti mistä tahansa 2D- kuvasta, tekstistä, kaaresta, logosta. Jopa skannattu kuva voidaan vektoroida ja muuttaa lopuksi STEP- formaatin malliksi ja sitä kautta robottiradaksi.

Etäohjelmointi on vielä aika uusi asia konepajojen robottisovelluksissa, vaikka onkin ollut jo pitkään käytössä auto- ja lentokoneteollisuudessa. Testiympäristön esittely Wärtsilän edustajille oli paljon ideoita antava, ja uskon että etäohjelmointiympäristöt tulevat käyttöön pienemmässäkin mittakaavassa. Työn tavoitteet saavutettiin, ja seuraava vaihe on konseptin käyttö sekä jatkokehitys.

## LÄHTEET

Black J.T., 1999. Handbook of Industrial Robotics, 2<sup>nd</sup> edition: Fabrication and Processing 831-866, John Wiley & Sons, Inc. USA, ISBN 0-471-17783-0

Bogue Robert, 2009. Industrial Finishing robots: a review of technologies and applications, An International Journal 36/1 (2009) 6–12 Emerald Group Publishing Limited [ISSN 0143-991X] [www.emeraldinsight.com/0143-991X.htm](http://www.emeraldinsight.com/0143-991X.htm)

David A. Stephenson, John S. Agapiou, 1997. Metal cutting theory and practice. Marcel Dekker, Inc, USA ISBN 0-8247-9579-2

Delfoi Oy. Tuotannon simulointi. [WWW]. [viitattu 28.12.2011].

Saatavissa: [http://delfoi.fi/web/solutions/production/fi\\_FI/tuotannon simulointi/](http://delfoi.fi/web/solutions/production/fi_FI/tuotannon simulointi/).

Fanuc Corporation, 2001. FANUC Robot series (R-30i A CONTROLLER) Operator's Manual, B-82594EN-2/01.

Fanuc Robotics, 2011. Products [WWW]. [viitattu 28.12.2011].

Saatavissa: <http://www.fanucrobotics.com/Products/Robots/Atoz.aspx>

Fastems Oy. Net Integrator. [WWW]. [viitattu 5.2.2012].

Saatavissa <http://www.e-fastems.com/magazine/index.php/2/11/FI/2011/01/fastsimu/>

Glaser Andrew, 2008. Industrial Robotics: How to implement the right system for your plant automation, Industrial Press Inc. New York, ISBN 978-0-8311-3358-0

Goldenberg Andrew A., Emami Mohammad R., 1999. Handbook of Industrial Robotics, 2<sup>nd</sup> edition: Kinematics and Dynamics of Robot Manipulators, John Wiley & Sons, Inc. USA, ISBN 0-471-17783-0

Goldenberg, A. A. and Emami, M. R., 2007 Kinematics and Dynamics of Robot Manipulators, in Handbook of Industrial Robotics, Second Edition (ed S. Y. Nof), John Wiley & Sons, Inc., Hoboken, NJ, USA. doi: 10.1002/9780470172506.ch6

Graig John J., Introduction to Robotics Mechanics and Control, Paerson education inc, 2005, 400 s, ISBN 0-13-123629-6

IFR, The International Federation of Robotics. [WWW]. [viitattu 5.4.2012].

Saatavissa: <http://www.ifr.org/industrial-robots/products/>

Kauppinen, V, 2003. Suurnopeusjyrsintä, Teknologiateollisuus ry, Helsinki.  
ISBN 951-817-820-8

Kim C. and Chung J. 2006. "Robust coordination control of a pneumatic deburring tool", Journal of Robotic Systems, Vol. 22, Supplement, pp. S1-S13.

Kuivanen Risto, 1999. Robotiikka, Suomen robotiikkayhdistys Ry Talentum Oyj Vantaa. ISBN 951-9438-59-0

Milton C. Shaw, 2005. Metal cutting principles. 2<sup>nd</sup> edition. Oxford University Press, New York, ISBN 0-19-514206-3

Mohsen M. Daniali and Gholamreza Vossoughi, 2009. Intelligent Active Vibration Control of Constrained Manipulators in Robotic Deburring, International Conference on Industrial Mechatronics and Automation 2009, ICIMA 2009, 15-16 May 2009, ISBN 978-1-4244-3818-1

Niku Saeed B. 2001. Introduction to Robotics Analysis, Systems, Applications Prentice Hall, Inc. New Jersey ISBN 0-13-061309-6

Odham Aaron, 2007. Tooling & Production Robotic Deburring: Making Choices, 12/2007 Nelson Publishing Inc. [www.toolingandproduction.com](http://www.toolingandproduction.com)

Pennock Gordon R. and Ryuh Beom-Sahng, 2006. Robot Automation Systems for Deburring, Industrial Robotics: Programming, Simulation and Applications, Low Kin Huat (Ed.), ISBN: 3-86611-286-6, InTech,

PushCorp Inc. Technical papers. [WWW]. [viitattu 5.2.2012].  
Saatavissa: <http://www.pushcorp.com/Tech%20Papers/Force-Control-Basics.pdf>

Robinson, Stewart, 2004. Simulation – The Practice of Model Development and Use. England: John Wiley & Sons Ltd.

Schröer Klaus, 1999. Handbook of Industrial Robotics, 2<sup>nd</sup> edition: Precision and Calibration 795-810, John Wiley & Sons, Inc. USA, ISBN 0-471-17783-0

SFS-EN ISO 10218-1 Teollisuusrobotit. Turvallisuusvaatimukset. Osa 1: Robotti

Siirilä Tapio 2009. Koneturvallisuus, Ohjausjärjestelmät ja turvalaitteet ISBN 978-951-98254-65

Teknologiaeollisuus. Trioplus-seminaari Alihankinta messuilla 2011. [WWW]. [viitattu 28.12.2011] Saatavissa:

<http://teknologiaeollisuus.fi/file/12007/SuuriKvaloppuraportti22.6.2011MDI.pdf.html>

Tolinski Michael, 2006 Deburring Processes and Challenges Manufacturing Engineering; Oct 2006; 137, 4; ABI/INFORM Complete pg. 83

Työsuojeluhallinto, Työsuojelujulkaisuja 91, 2009

Vamk Oy. Vaasan ammattikorkeakoulun verkkosivut. [WWW]. [viitattu 1.4.2012].

Saatavissa <http://www.puv.fi/fi/>

Vuorinen Ilpo, 1993. Tuhat tapaa opettaa. Vammala: Vammalan Kirjapaino Oy.

Weilercorp Inc. Technical papers. [WWW]. [viitattu 5.2.2012].

Saatavissa: <http://www.weilercorp.com/nafttech.htm>

Yong Liu and Ning Xi, 2011. Industrial Robot: An International Journal 6/2011: Joint offset of industrial robot using single-point constraint, Emerald Group Publishing Limited ISSN 0143-991X